Malware Analysis Quant: Phase 1 – The Process

Version 1.2

Released: January 30, 2012

Securosis, L.L.C. 515 E. Carefree Blvd. Suite #766 Phoenix, AZ 85085 **T** 602-412-3051 info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog, but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Licensed by Sourcefire

Sourcefire, Inc. (Nasdaq:FIRE), a world leader in intelligent cybersecurity solutions, is transforming the way global large- to mid-size organizations and government agencies manage and minimize network security risks. With solutions from a next-generation network security platform to advanced malware protection, Sourcefire provides customers with Agile Security™ that is as dynamic as the real world it protects and the attackers against which it defends. Trusted for more than 10 years, Sourcefire has been consistently recognized for its innovation and industry leadership with numerous patents, world-class research, and award-winning technology. Today, the name Sourcefire has grown synonymous with innovation, security intelligence and agile end-to-end security protection. For more information about Sourcefire, please visit www.sourcefire.com.

Copyright

This report is licensed under Creative Commons Attribution-Noncommercial-No Derivative Works 3.0.

http://creativecommons.org/licenses/by-nc-nd/3.0/us/

Table of Contents

The Malware Analysis Process 4

Confirm Infection Subprocess 5

Analyze Malware Subprocess 6

Malware Proliferation Subprocess 7

Confirm Infection 8

Notification 8

Quarantine 9

Triage 10

Confirm 11

Analyze Malware 12

Build Testbed 12

Static Analysis 14

Dynamic Analysis 16

The Malware Profile 19

Malware Proliferation 21

Define Rules 21

Find Infected Devices 22

Remediate 24

Monitor for Reinfection 25

Coming Soon: Phase 2 27

About the Analyst 28

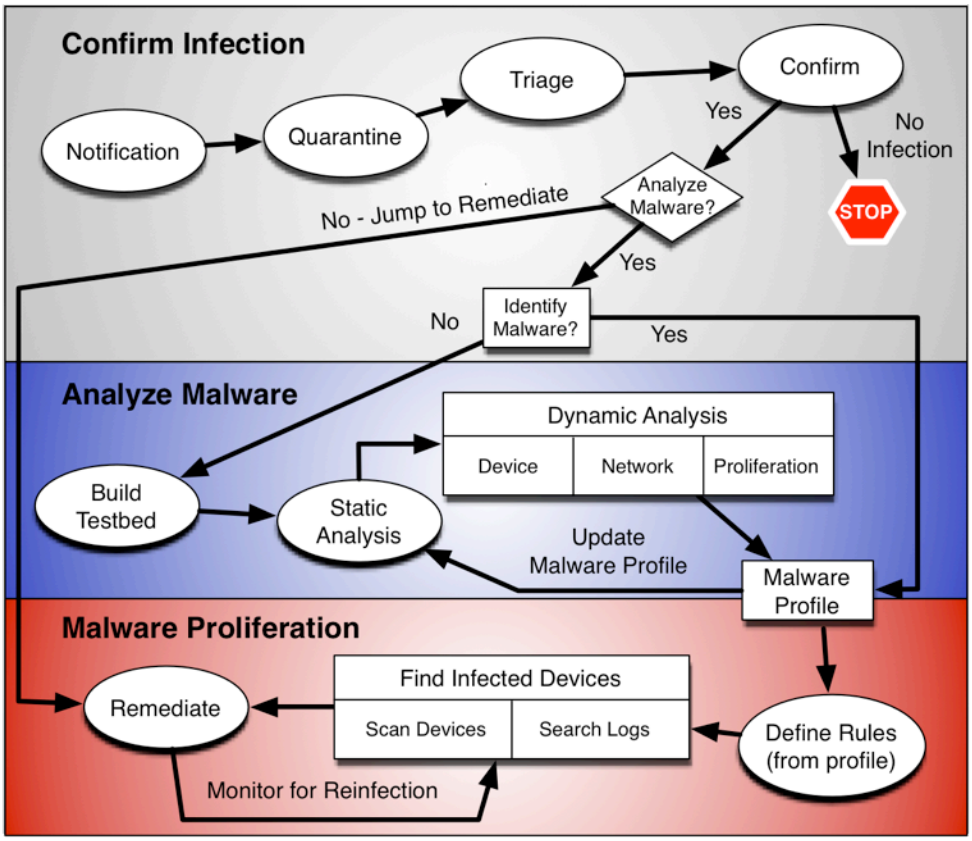
About Securosis 29

The Malware Analysis Process

Developing an Open Malware Analysis Metrics Model

We launched the Malware Analysis Quant research project — MAQ for short — to develop an unbiased metrics model to describe the costs of confirming, analyzing, and assessing the damage caused by malware infection. We want to provide organizations with a tool to better understand the security costs of these activities, both the things they do and the things they don't. By capturing quantifiable and precise metrics that describe the daily activities of security practitioners, malware analysts, and incident responders we can better understand the costs associated with security and compliance efforts. Malware Analysis Quant has been developed through independent research and community involvement to accurately reflect all substantive efforts that comprise a malware containment program.

We start each Quant project with a detailed process map. Then we describe the processes in gory detail. Later in the project we assign metrics and the costs for each step in the process.



To provide a brief description of what me mean by "Malware Analysis" this project addresses how organizations confirm, analyze, and then address malware infections. This is important because today's anti-malware defenses basically don't work (hard to argue) — so as a result way too much malware makes it through defenses. When you get infected you start a process to figure out what happened. First you need to figure out what the attack is, how it works, how to stop or work around it, and how far it has spread within your organization. That's all before you can even think about *fixing* anything. So let's jump in with both feet.

Confirm Infection Subprocess

This process typically starts when the help desk gets a call. How can they confirm a device has been infected?

1. **Notification:** The process can start in a number of ways, including a help desk call, an alert from a third party (such as a payment processor or law enforcement), or an alert from an endpoint suite. However it starts, you need to figure out whether it's a real issue.

2. **Quarantine:** The initial goal is to contain the damage, so the first step is typically to remove the device from the network to prevent it from replicating or pivoting.

3. **Triage:** With the device off the net, now you have a chance to figure out how sick it is. This involves all sorts of quick and dirty analysis to figure out whether it's a serious problem – exactly what it is can wait.

4. **Confirm:** At this point you should have enough information to know whether the device is infected and by what. Now you have to decide what to do next.

Based on what you found you will either: 1) stop the process (if the device isn't infected) or 2) decide whether to analyze the malware or just remediate the device. If you decide to analyze the malware, 3) analyze the malware (if you have no idea what it is), or 4) assess malware proliferation (if you know what it is and have a profile).

To be clear, it doesn't make a lot of sense to us to totally skip over any form of malware analysis and just remediate the device and move on. At a minimum, you should be looking for other infected devices in your environment based on what you learned during this confirmation subprocess. But again, our job in this research is to present a complete process map, and that means we have to factor in the reality that some organizations don't do any type of analysis.

Analyze Malware Subprocess

By now you know there is an infection but not what it is. Is it just an annoyance or is it stealing key data and presenting a clear and present danger to the organization? Here are some typical malware analysis steps for building a detailed profile.

1. **Build Testbed:** It's rarely a good idea to analyze malware on production devices connected to production networks. So your first step is to build a testbed to analyze what you found. This tends to be a one-time effort but you will always be adding to the testbed based on the evolution of your attack surface.

2. **Static Analysis:** The first actual analysis step is static analysis of the malware file to identify things like packers, compile dates, and functions used by the program.

3. **Dynamic Analysis:** There are three aspects of what we call Dynamic Analysis: device analysis, network analysis, and proliferation analysis. To dig a layer deeper, first we look at the impact of the malware on the specific device, dynamically analyzing the program to figure out what it actually does. Here you are seeking perspective on the memory, configuration, persistence, new executables, etc. involved in execution of the program. This is done by running the malware in a sandbox. After understanding what the malware does to a device you can start to figure out the communications paths it uses. This includes command and control traffic, DNS tactics, exfiltration paths, network traffic patterns, and other clues to identify the attack. Finally you'll need to understand whether and how the malware spreads, which we call proliferation analysis. You look at the kind of reconnaissance it performs, along with any other clues that indicate the malware is running rampant in your environment.

4. **The Malware Profile:** Finally we need to document what we learned during our malware analysis, packaged up into what we call a Malware Profile.

With a malware profile in our hot little hands we need to figure out how widely it spread. That's the next process.

Malware Proliferation Subprocess

Now you know what the malware does, you need to figure out whether it's spreading, and how much. This involves:

1. **Define Rules:** Take your malware profile and turn it into something you can search on with the tools at your disposal. This might involve configuring vulnerability scan attributes, IDS/IPS rules, asset management queries, etc.

2. **Find Infected Devices:** Then take your rules and use them to try to find badness in your environment. This typically entails two separate functions: first run a vulnerability and/or configuration scan on all devices, then search logs for indicators defined in the Malware Profile. If you find matching files or configuration settings you need to be alerted to another compromised device. Then search the logs, as malware may be able to hide itself from a traditional vulnerability scan but might not be able to hide its presence from log files. Of course this assumes you are actually externalizing device logs. Likewise you may be able to pinpoint specific traffic patterns that indicate compromised devices, so look through your network traffic logs, which might include flow records or even full packet capture streams.

3. **Remediate:** Finally you need to figure out whether you are going to remediate the malware (via reimaging or cleaning the device), and if so how. Can your endpoint agent clean it? Do you need to reimage? Obviously the cost of cleanup which must be weighed against the likelihood of reinfection.

4. **Monitor for Reinfection:** One of the biggest issues in the fight against malware is reinfection. It's not like we are dealing with static attacks. Malware changes constantly – especially targeted malware. Additionally, some of your users might make the same mistake and become infected with the same attack. Right, oh joy, but it happens – a lot. So making sure you update the malware profile as needed, and checking continuously for new infections, are key parts of the process as well.

Confirm Infection

Let's start by considering the subprocesses around confirming infection – always the first step. Obviously until you know there's a problem there's nothing to analyze. These steps are pretty straightforward, as we described in the process map:

1. **Notification:** The process can start in a number of ways, including a help desk call, an alert from a third party (payment processor, law enforcement, etc.), or an endpoint suite alert. However it starts you need to figure out whether it's a real issue.

2. **Quarantine:** The initial goal is to contain the damage, so the first step is typically to remove the device from the network to prevent it from replicating or pivoting (jumping to another device on your network).

3. **Triage:** With the device off the grid, now you get to figure out how sick it is. This step involves all sorts of quick and dirty analysis – it's not about figuring out exactly what it is, but simply whether it's a problem.

4. **Confirm:** At this point you should have enough information to know whether the device is infected and by what. Now you have a decision to make: what to do next.

Based on what you find you might: 1) stop the process (if the device isn't infected), 2) analyze the malware (if you have no idea what it is), or 3) assess malware proliferation (if you know what it is and have a profile).

Notification

The notification step kicks off the entire process and usually involves kicking off a structured process to figure out whether the device is infected. This involves the following distinct subprocesses:

1. **Fact Finding:** When something is suspicious the first step is always to figure out what's going on, which usually involves asking a bunch of questions. You need to figure out why someone thinks the device is infected. Is its performance bad? Did the user click something they shouldn't have? Did law enforcement find your secret sauce on a black market site? Has your payment processor reported a rash of fraudulent transactions which they traced back to you? There are a few sets of likely starting points, and you should have a structured questionnaire for each one because the first-line defenders (those performing this initial fact-finding) won't be sophisticated malware analysts. So give them a script to ensure the right information is captured the first time.

2. **Find devices:** The next step is to actually find the devices in question. This sounds trivial, but to take quick action (such as capturing an image for forensics) you need to know where the device is. Is it a mobile device? Is it connected to your network? You'll want to consult your CMDB (configuration management database) and network maps to determine whether you can isolate the machine quickly.

3. **Escalate:** Now it's time to get the second line of defense involved. These usually aren't help desk operators, but instead network and/or system admins who will take the first active steps in investigating potential infection. Again, we recommend documenting all these hand-offs and escalations ahead of time – it is essential that you avoid uncertainty regarding roles and responsibilities when dealing with a potentially rapid infection.

Before moving on to the next subprocesses it is worth reiterating the importance of structure – for both the fact-finding and escalation aspects of this step. You don't want your specialized (and expensive) malware analysts to spend a bunch of time asking simple questions again. So make sure the folks answering the phones or fielding the initial requests, know what questions to ask and who to call based on the answers they get.

Through this research we continually highlight the need to practice the response process frequently. The bad news is that you'll likely have plenty of opportunities to analyze real malware infections — such is the nature of being a security professional. But practice is an important part of a comprehensive process model. We always say you don't want to find holes in your response process *during* a real response.

Quarantine

Job #1 of any incident response activity is to contain the damage. You want to remove any questionable devices from the network as quickly as possible, and prepare to analyze them. That's what the quarantine subprocess is all about.

1. **Isolate devices:** First and foremost prevent devices from spreading infection, so remove them from the network as quickly as possible. That may mean shutting down network ports with the network ops team, locking them out of the network if they are offline or mobile, or unplugging them from the network and turning off other communications technologies like Bluetooth. Remember finding the devices during the notification subprocess? Now that information is essential for getting them off the network quickly. You don't yet know the nature of the attack or its persistence, so don't turn machines off or on, or really mess with them yet.

2. **Image devices:** At this point you don't know the nature of the infection or what is at risk, so quickly take a forensic image of the machine. There may be foul play involved, and you can't assume law enforcement won't be, so the faster you capture the image the better. Obviously this can be challenging for devices not in your physical control, but do the best you can – until you get physical access to the device your ability to confirm the infection is limited.

We know end users get grumpy when you take their machines, especially if they might lose data. Oh, well, it's a chance to exercise your empathy. But you need to progress to the next step, which means the user needs to figure something else out.

Triage

Just like the medical kind, malware triage is about figuring out how sick a device is. There are many ways to do it, so let's map out the typical steps in this subprocess. Of course, depending on the nature of the potential infection, you might skip some of these steps. Or you might do others, but the point is to get a quick and dirty analysis of what happened. You don't need conclusive answers yet but should be able to make an educated decision about how significant the infection is, if there is one.

1. **Observe behavior:** Perhaps the user said the device is slow. Can you confirm it? Is it throwing pop-ups or other clear indications of malware? This is the obvious stuff.

2. **Run AV scan:** Okay, stop laughing. But part of this process is to actually run an AV scan to figure out whether it's something obvious and well-known. We all accept that traditional AV isn't going to catch anything really novel, but checking for low-hanging fruit is one step in the triage process. And if the AV scan does match at least you know what you're dealing with.

3. **Check configuration/registry:** Another area to check is the device configuration and/or registry, to pinpoint changes that might indicate compromise. Of course it's helpful to be able to pull a diff to compare its configuration and registry against a known good reference, because you might find a clear indication here of what changed.

4. **Investigate open processes/threads:** You can also check the open processes and threads on the device to pinpoint possible malware. Some modern malware does a good job of compromising a trusted process but there are often clues here.

5. **Examine file activity:** If malware has written new files or replaced existing ones you'll want to know what happened, so check out the device logs or use a forensics tool to look at file activity timelines. When trying to identify the type and severity of the infection you will want to identify the involved files so this is critical.

6. **Analyze memory:** You will also want to check out a memory dump from the device for problems. Yes, you will need a sophisticated tool for this, but often the only traces of an infection are in memory so you can't neglect this check.

Of course there are forensics tools to automate many of these steps. But Quant process maps describe, at a granular level, what someone would do manually, in order to figure out the cost/benefit impact of various levels of automation. You may be better off just using the SANS SIFT toolkit, Mandiant's Redline and IOC Finder, or the various other available tools, rather than trying to figure this all out manually. But these maps cover what the tools do to provide an accurate idea of the effort required for each subprocess.

Ultimately, coming out of the triage subprocesses, you should have enough information to know what you are dealing with, and then you can decide the next step.

Confirm

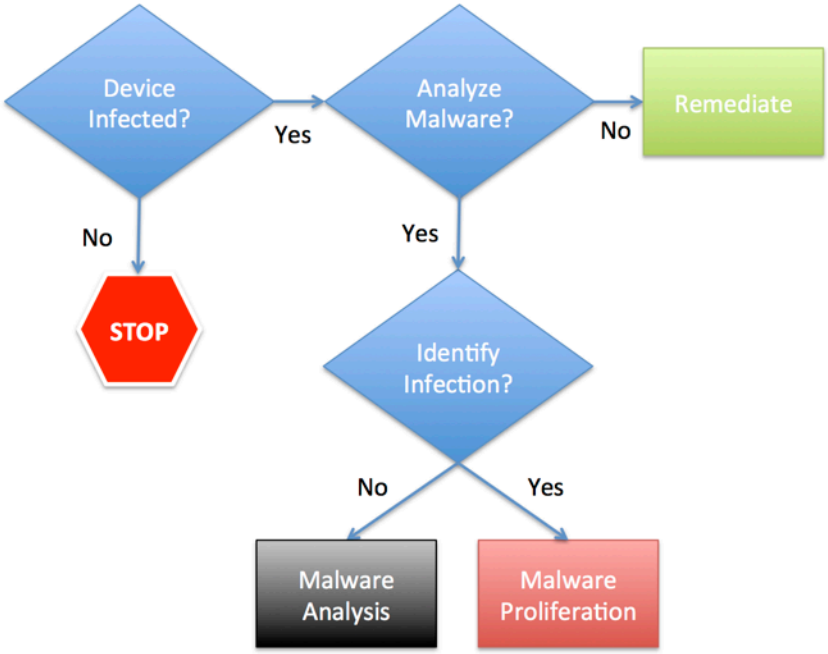
It's decision time. Is the device infected? If so, do you know by what? What's the next step? In this subprocess you need to answer all these questions and determine the next step.

1. **Infection decision:** From triage you should know definitively whether the device is infected. If not you are done – great. If it is then continue to the next step.

2. **Analyze malware decision:** If you know the device is infected, then you have to decide whether you want to analyze the malware. By analyze, we are referring to either the detailed analysis to profile the malware or even just to see if there are other devices infected with similar malware already in your environment. As we mentioned above, we don't recommend organizations just remediate a device and move on, but it happens, so we've factored that into the process map.

3. **Infection identification:** The device is infected and you've decided to do some type of analysis on the malware. Next you need to figure out whether you know what the malware is. If so, you have been able to identify the specific attack (maybe via your AV scan <snicker>, or via a file or memory string) and can access a profile of what the attack does and more importantly how to find it on any other devices on your network. If that's the case, you skip malware analysis and jump directly to the malware proliferation activities. If you don't know what the infection is, then your next step is the analyze malware activities.

So your decision tree looks like this:



Based on this decision tree you either go directly to the Remediate step, proceed to the Analyze Malware activities, or start to figure out how widely the infection has spread (Malware Proliferation).

Analyze Malware

You know there is an infection but you not yet what it is. Is it just an annoyance, or is it stealing critical data and presenting a clear and present danger to the organization? The next subprocess digs into the malware file(s) using a number of different techniques. The output of this step is a profile of the malware, which will provide the basis for searching your environment during the Malware Proliferation subprocess.

Of course (as with all our Quant research) not every company does all the steps laid out in this process map. In fact most organizations don't do many of these functions — instead relying on their security vendors or third party incident responders for detailed malware analysis. Some organizations don't do any malware analysis at all. They just blow the infected device away and hope they don't get infected again. We don't judge right or wrong. Of course we have opinions but the objective of a Quant project is to educate readers about all the things that can be done and provide enough information for you to decide what makes sense for your environment.

Build Testbed

The first step in the Malware Analysis process is to set up a testbed because you need somewhere to play. This is mainly done once, but keeping your environment and tools current require some ongoing effort. For now let's just say a bit about the environment you should strive to build.

Build Testbed: It's rarely a good idea to analyze malware on production devices connected to production networks. So your first step is to build a testbed to analyze what you found. This tends to be a one-time effort but you will always be adding to the testbed based on the evolution of your attack surface.

Of course we need to mention caveats. Some of you will not need all of this stuff, while others will need more. There is no one-size-fits-all test environment but we have some guidelines and general tool categories you will need.

- **Isolated test network:** Don't test live malware on production networks, so you'll need a separate network to host your testbed.
- **Victim devices:** You need victim machines in a variety of configuration states (patched, not patched, etc.) and a variety of operating systems from your environment. You will want to rely heavily on virtualization snapshots and re-imaging tools to wipe and rebuild victims

quickly. Given that many malware writers check whether their malware is running in a virtual machine, you will need some physical machines as well (how 2005!) to test on.

- **Network services:** You should probably provide a totally separate Internet connection — which doesn't have to be fast. At minimum you need DNS and outbound Internet connectivity, which you could simulate on a closed network (a network not connected to anything else), using tools like Joe Stewart's Truman tool). But from our research providing direct Internet connectivity is preferred – you likely want to use a physically separate network provided by a separate ISP to make sure there is no way a compromised machine in the testbed can jump to the production network.

- **Testing tools:** There are many things you can buy, as well as a lot of open source and shareware. Lab tools are a personal preference so you will likely need to try a bunch of stuff before you find a set of tools that work for you.

  - **Imaging tools:** Before you start investigating you need a clean image for forensics and possibly prosecution.

  - **File/data analysis:** You will want some tools for static analysis of potential malware files – including things like disassemblers, decompilers, and source code analyzers. Be sure your tools can provide a timeline of which files are moved, added, and changed, as that's key to understanding what the malware does.

  - **Registry/configuration analysis:** Most malware targets Windows and messes with the registry and other configuration variables, so a tool to run quick diffs against last-known-good or gold master settings can save a lot of time.

  - **Sandbox:** You can do dynamic analysis manually, but it's not for the feint of heart, and it may not make sense when there are both self-contained sandbox appliances and services that can help analyze executables. I'm using this generic term to also include memory analyzers and the like which allow you to build your own sandbox.

  - **Log analyzers:** Devices under attack generate log files recording activities of malware, so you need some kind of log aggregator and parser to wade through what could be thousands of log records.

  - **Network capture:** You will also need to understand how the malware leverages the network, so you need to capture traffic to isolate command and control streams, possible exfiltration paths, and encryption mechanisms.

Of course not everyone needs all these tools, but this should be a reasonably comprehensive list of things you might need to get your testbed up and running. Now that we have the testbed in place we can start to analyze bad stuff.

Static Analysis

Now it's time to get busy and actually start analyzing a suspected file. First let's spend a minute talking about the theory behind starting your file analysis. The output of this analysis is a profile of the malware attack, so you can search for the malware in your environment. Malware writers leave plenty of clues within their executable files, which can be used to identify future infections. The steps include:

> **Static Analysis:** The first of the actual analysis steps involves a static analysis of the malware file to identify things like packers, compile dates, and functions used by the program.

1. **First pass static analysis:** Look for obvious indications of what the attack is and whether you have seen it before. Basically you check the file fingerprint (usually a MD5 hash) and make sure it doesn't match known malware. A few services maintain extensive cloud-based file repositories which can tell you whether any file you find is a match. If this sounds like traditional AV, that's because it is. But if the AV vendors have already seen it, shame on you if you don't quickly identify it.

2. **Deep dive file analysis:** If you don't get a match on the fingerprint it's time to look deeper and figure out what you can about the file. Here you use tools and techniques such as:

   - **File packing:** File packing is the function of compressing a file, usually to save space. But attackers use packing as a means to obscure its content and/or behavior, so you need to figure out if the packing process potentially hides some badness since file packing blocks most other techniques of static analysis. By identifying the packer to see if it's prominently used by the malware writers and then trying to unpack the file, you can figure out how effective the rest of the static analysis will be.

   - **File classification:** If you are analyzing Windows malware (which is most of it), the executable has a file structure (called Portable Executable) that may yield information about the attacker and their technical capabilities. Things like compile information, version info, menus, function calls, etc., can all be leveraged by an experienced malware analyst to pinpoint a probable adversary. But of course this kind of easy marker can be used for disinformation as well.

   - **Plain text matching:** Assuming the file wasn't packed or was successfully unpacked, you can isolate text strings embedded in the executable that might indicate something about who wrote the file, where it communicates to, or how it works. Search engines are your friends here, as anything interesting can be queried to figure out whether someone else has already found your particular string.

   - **Disassembly:** The last static analysis technique is disassembly: using a tool like IDA Pro to examine the machine code of the executable and step through it as in a debugger to figure out exactly what the program is doing. This is pretty advanced, but if you want to figure out what the malware does you need to get into actual program execution.

Of course this simplified description of the static analysis process doesn't go into specifics. There are many books, training courses, and other resources to consult when you want to perform an actual static analysis.

As we have said many times, there is no panacea – figuring out what the malware does is detective work. And of course malware writers don't make this easy – they tend to obscure their attacks by packing, encrypting, and otherwise hiding the attack code within a lot of irrelevant content. Compared to dynamic analysis, static analysis is pretty safe because you don't execute the dangerous code – unless you totally mess up. But it's still best to undertake static analysis on an isolated machine to contain any mistakes.

Do you need to perform static analysis? No, but otherwise you can waste a lot of time on malware which has already been identified. As with everything, what makes sense for *your* environment varies. Some folks skip right into dynamic analysis but our research shows that checking out the file statically is generally worthwhile.

Dynamic Analysis

As we described above, malware analysis typically starts with a static analysis of the file before you let the malware run in the lab. But for most folks the fun starts when you run the bad stuff to figure out what it does. We start by running the malware in your testbed. Each aspect of this analysis involves looking at different type of data captured when the malware runs. As we discussed when building the testbed, you should only run live malware in isolated environments. You don't know what the malware will do yet, so be careful.

Keep the goal in mind: To develop a profile of the malware which allows you to find other devices in your environment that are compromised, and establish rules to make sure the same attack isn't successful in the future. As fun as it is to figure out what malware does, and parse the innovative new techniques attackers use to evade detection, the point is to figure out the extent of the damage to your environment and how to stop it.

Device Analysis

Malware typically changes all sorts of things on the compromised device so that's where we will begin dynamic analysis. Let's look at the type of information to gather and why.

> **Device Analysis:** First look at the impact of the malware on the specific device, dynamically analyzing the program to figure out what it actually does. Here you are looking to gain perspective on the memory, configuration, persistence, new executables, etc. involved in execution of the program. This is done by running the malware in a sandbox.

- **Volatile Memory:** Malware can overflow buffers and/or tamper with program memory to gain access to device. By capturing and then analyzing the device memory you can figure out whether and how the malware uses memory.
- **Configuration/Registry Changes:** Look for any evidence of the malware changing configurations and registry keys. You are exposing it to a victim machine in a known (vulnerable) state so you know exactly what the starting configuration/registry looks like and can easily isolate changes the malware makes.
- **File Activity:** Malware also may add, change, delete, or otherwise tamper with files. So you should have a log of file activity to pinpoint what the malware changes when it runs against the victim device. Checking the hashes of new and/or changed files can provide a lot of information on what the malware does. You'll want to leverage access to the malware database used during static analysis to check these new files against known malware hashes as well, because even a zero-day targeted attack may launch familiar files after gaining access to a device.
- **Processes/Services:** Also look for new or stopped processes and/or services. A lot of malware shuts down AV engines, or even starts one to remove or block competitive malware; or it might add kernel-level devices to sniff the network or anything else the authors could think of. As with the configuration/registry

analysis, you'll know what processes/services were running when you set up the victim device so you can quickly isolate what the malware changed.

- **Persistence:** You can also figure out what (if anything) the malware does to run again on restart. Perhaps it's a root kit, or the aforementioned registry changes, or an executable in a startup folder. Regardless of technique — by analyzing changes to the configuration, registry, and files — you should be able to figure out whether or not, and how, the malware attempts to restart itself.

- **VM Awareness:** A lot can be said about malware writers but they aren't dummies. Many now build in tests to figure out whether their malware is running inside a VM. On detecting virtual machines, malware tends to go dormant to avoid detection on virtualized detection platforms. So you'll also need physical devices running your vulnerable build directly on hardware to deal with VM-aware malware.

Clearly this analysis generates a tremendous amount of data – especially if you look at device logs and process monitors. So you'll need some reasonably tight filters as you start your analysis. You can always loosen up the filters to look at more data if you feel something is missing. But it's easy to be overwhelmed when looking at all the data.

Network Analysis

As we discussed it's hard for malware to effectively mask or obfuscate its network access. For one thing, most malware takes commands from some kind of command and control (C&C) network and eventually needs to exfiltrate the data. So network traffic analysis is essential to understanding and eventually profiling what the malware does. As before we start by running the malware to analyze its network traffic.

> **Network Analysis:** After understanding what the malware does to a device, you can begin to figure out its communications paths. This includes command and control traffic, DNS tactics, exfiltration paths, network traffic patterns, and other clues to identify the attack.

- **Capture Network Traffic:** Without the network traffic there isn't much to analyze. But first you have a key decision to make. You can capture the traffic right off the wire using a tool like WireShark or install a network capture driver on the victim machine to pull traffic directly off the device — or both. The advantage of pulling traffic directly off the device is that you can enrich the capture with information about specific processes to pinpoint potential executables or services that originate traffic. You also have to figure out, as part of building your testbed, whether you will simulate network services (using a tool like Joe Stewart's Truman to build a "sandnet") or provide the malware with real but restricted network access.

- **Look for Suspicious Destinations:** Once you have the traffic look for suspicious destinations – such as known command and control networks, compromised sites, etc. Many organizations grow their suspicious

lists organically, although commercial information services can provide IP reputation databases to fill out the list of IPs to look for.

- **Analyze C&C Traffic:** Once you have isolated traffic going to a known bad site you can analyze it to figure out how the bot master is communicating with compromised devices. The goal of this entire process is to profile the malware and be able to find other compromised devices, so being able to build IDS/IPS and/or firewall egress rules to sniff out C&C traffic is critical for detecting reinfection.

There are specific risks worth keeping in mind as you analyze network traffic, particularly remaining anonymous to the attackers. Increasingly the initial malware attack acts as a placeholder, from which the controller figures out what they want to do with their new asset. In many cases controllers keep a tight reign on what they are attacking, and if an IP address and/or device they didn't attack phones home they might discover something is up. It's like when the undercover agent is discovered in a mob movie – it never ends well for the agent. The controllers may decide to attack the originating network with a denial of service attack, or just go quiet for a while to stymie further analysis.

This underscores the importance of using an isolated network when testing malware. Isolation isn't necessarily merely a different IP address – it could entail using a separate Internet Service Provider or even geographic egress point. Don't irritate or panic attackers into taking down your production network. Nobody wins in that scenario.

Proliferation Analysis

We use the term "proliferation analysis" for the effort to understand how the malware spreads. Does it scan the network where it's run for vulnerable devices? That's pretty old school. Does it phone home and wait for a human controller to connect and do some analysis? Does it just wait for commands from the bot master, without any initial effort to spread internally? As usual we start the proliferation analysis by running the malware. Yes, you can run the malware once and capture all the data for the device, network, and proliferation analyses concurrently.

> **Proliferation Analysis:** Finally you will need to understand whether and how the malware spreads, what kind of reconnaissance it performs, along with any other clues to whether and how widely the malware is running rampant in your environment.

- **Set up Another Victim:** It's hard to figure out how malware spreads if it doesn't have a target to compromise, so set up another victim device for the malware to attack.

- **Capture Network Traffic:** Our network capture comes in handy again, as we use this information to pinpoint whether and when the malware starts scanning its vicinity for other targets. You will be able to see how the malware scans, what it looks for, and then what happens when it finds something of interest: your sacrificial victim.

- **Observe:** The proliferation analysis is done mostly through direct observation, rather than log and/or device analysis. Sure, you start by analyzing the network capture to figure out whether scans are occurring, but then you observe. Automated reconnaissance is pretty straightforward – it gets interesting when a human controller connects to the device. You will want detailed notes about what happens on the device – hopefully including information you can build into the profile, which we will talk about in later.

Do You Need to Perform Dynamic Analysis?

As with all the other steps, we need to at least *ask* whether you really need to undertake device analysis. The answer is a resounding *yes*. It's not clear how else you could develop a malware profile without this analysis. There are services that can analyze your files and send back a report of what they find, but the Quant model models the costs of performing the actual analysis, so this step isn't something we can skip over.

The Malware Profile

We wrap up the Malware Analysis subprocess by leveraging all the analysis done in the previous couple steps (Static Analysis and Dynamic Analysis), and building a profile to embody what we know about the malware attack.

The key for this step is *specificity*. The more work you do now to describe the malware, the easier it will be later to build rules which achieve your goals. So part of the analysis is digging deep, figuring out exactly what the malware does and identifying markers which will help find it. You need to describe those markers now, in language useful to the folks who build the rules to find the malware.

So packaging your malware profile looks like this:

1. **Aggregate findings:** This first step is to take all the information from your analysis (including the device, network, and proliferation analyses) and put it together in one place. Depending on the size of your malware analysis team you may pull from a bunch of different places. Here is a short (and not comprehensive) list of information types you might have.
    - File attributes
    - Registry settings
    - Processes/services
    - New executables
    - Domains/protocols

We need to keep our goals in mind when building the profile.

- **Assessing Malware Proliferation:** There are times when only one device gets infected during a malware attack. But other times it becomes an outbreak. So your first job after developing the profile is to figure out whether the malware has spread. That is our third subprocess, which we will dig into next week.
- **Preventing reinfection:** The other purpose of the malware profile is to make sure you don't get reinfected. We will dig into how later, as well.

- Command and control obfuscation
- Persistence/VM awareness

2. **Package Profile:** Document what you found in a way the folks looking for malware can leverage. If there is separation between malware analysts and the incident responders who look for infections then you need to work out the preferred packaging for this information. According to our research, the closer analysts can get to packaged rules which responders can just plug into their scanners and forensics tools, the better they will work together.

3. **Distribute Profile:** Depending on the size of your security team, there may be a number of folks who need access to the profile. Their interactions must be defined at the start of the process. Some organizations also share their analyses with key strategic vendors, industry information sharing groups, or mailing lists. So your profile might also be used externally, which may impact the type and depth of documentation you produce.

Typically we don't highlight vendor activity in process maps, but we need to mention the work Mandiant has done with the OpenIOC initiative. They have produced a set of XML schemas which describe the types of information necessary to identify and find malware in your organization, and provided them as open source. Of course this is self serving – Mandiant's incident response tools leverage the formats, so the more broadly OpenIOC is adopted the better for them, but we haven't found another comprehensive set of descriptors for malware indicators.

Revisiting the Malware Profile

The only thing we can count on from malware writers is that they will not stand still. They continue to adapt and evolve their malware to avoid detection, to increase its infection rate, and sometimes to add control features to better leverage infected systems. So we need to revisit malware profiles periodically, looking for changes in their indicators. How often will you revisit the profile? Basically every time you find the malware in your environment, as there might be new or changed indicators that require an update to the profile, and reinfection implies you need to update.

We also recommend that, if you can identify the name of the malware once anti-malware vendors have profiled and named it, you watch malware lists and other information sources for new information about it. For each of the high-profile attacks (ZeuS and Stuxnet come to mind), you will notice that the research community continues to find different variants, which means you need to update your profile.

With a very detailed (and preferably technical) profile, the incident response team can try to figure out whether, and if so and how badly, the malware has spread throughout your organization.

Malware Proliferation

Now we can decompose the final subprocess: figuring out how badly your organization has been infected. This next subprocess builds on everything you have done to date, leveraging the analyses to determine how badly your organization has been compromised.

Define Rules

The first step in this Malware Proliferation subprocess is to define the rules you will use to find the malware with the tools you have.

It looks like this:

> **Define Rules:** Take your malware profile and turn it into something you can search on with the tools at your disposal. This might involve configuring vulnerability scan attributes, IDS/IPS rules, asset management queries, etc.

1. **Develop rule:** First develop the rule. Sorry, but we have to be pedantic for Quant. This depends heavily on the tool you will use to (try to) isolate infected devices. For instance you might need to build a custom rule for your vulnerability scanner. Maybe you will build an IDS rule to look for command and control targets in your egress traffic. Perhaps you will search your CMDB for specific configuration/registry settings or executables. Most likely all of the above.

2. **Test rule:** Be happy you have a testbed environment – now you get to test your shiny new rule. That means actually infecting a victim machine (in an isolated environment of course) and seeing whether the rule works. If so, move on to the Document step. If not, figure out what needs to be changed and *then* move on to the next step.

3. **Refine rule (and retest):** After failing the test (or perhaps just not exactly passing), make the appropriate changes and try again. Depending on how complicated the malware is this might involve a few rules (typically 3-4) or many. And if you have sophisticated malware analysts on staff you might not need to define as many rules, as analysts can confirm other indicators defined in the profile without using other tools to confirm.

4. **Document rule:** Once the rule is tested and passes muster you need to document what it looks like. Again, how formally you document the rule(s) depends on how many different groups you have involved in incident response. If it's a small team you might be able to get by with streamlined documentation. But

for large teams, particularly if third parties are involved, you need to be fairly formal with documentation. Especially if a distinct operations group will to run the scans or set up the rules on devices they control.

5. **Go back to step 1 for the next indicator:** As mentioned above, it's pretty rare for there to be one smoking gun indicator that enables a simple rule to identify malware and determine proliferation. Once you finish building a rule based on a specific indicator, go back to Step One and start building the next rule, based on the next indicators in the profile. Remember, the tighter you define your search criteria, the less false positives will waste your time and money.

Finding ZeuS

To see how this process works let's take a look at the ZeuS malware. You can find its attack profile on the OpenIOC site, and if you parse the XML you will see a few indicators that identify this particular attack. Without going through *all* the indicators you can quickly see a number of process indicators which describe the processes ZeuS tends to use. You can scan all your vulnerable devices for these processes.

If you want to look for the specific network sites typically associated with ZeuS, you could consider the approach documented on Sourcefire's VRTLabs site. They reference the cool ZeuS Tracker, which lists the C&C servers and fake URLs it uses.

We generally have a decent amount of information available on how to find the widespread attacks within our environments. That also means you'll need to figure out the best approach for tracking proliferation. Depending on the attack you might want to run tests in a different order or skip certain tests entirely. Finding malware tends to be a very particular endeavor, and that makes it, uh, 'fun'. If you're into that kind of thing.

Find Infected Devices

Now we get to actually do something and look for a 'smoking gun'. Here we will use testing tools and log analysis to pinpoint infections.

Scan Devices

Let's start with testing tools. Here are the steps involved in scanning devices to find infections:

1. **Deploy rule on testing tool:** This may be a scanner, pen testing tool, configuration manager, forensics tool, etc. You need to generate a rule for your tool from the profile developed in the previous step.

2. **Run tool:** Run the rule on the tool. We know this is obvious – it's just part of laying out the whole process in sufficient detail for Quant and part of the cost model. So we have to list everything.

3. **Analyze results:** Once the tool finishes you analyze the results. Maybe a number of devices have clearly been compromised. Perhaps it's less obvious and you need to start looking for other markers. Either way you need to wade through the results to determine which devices have in fact been compromised.

4. **Document results:** If you are performing the analysis and scanning then the documentation may be as simple as a device name or IP address on the back of a napkin. But if you have many hands in the

process, with separate groups responsible for response and remediation, your documentation will need to be a bit more formal. Don't assume the operations team (or whoever is responsible for remediation) has any background on this type of malware; don't assume anything about the impact of the attack; don't assume everybody feels the full urgency of fixing it; don't assume anything. Everything must be spelled out.

5. **If infected, proceed to remediation:** If there is a clear sign of infection then continue to remediate, leveraging your top-notch documentation. Obviously remediation entails many decisions, but let's not put the cart ahead of the horse. We will get into that once we have finished looking for indicators of an attack.

Search Logs

Searching logs is very similar to running testing tools. So let's not belabor the point, and just jump right in:

1. **Search logs:** Similar to the 'rules' used by the testing tools, search your logs for the indications defined in the malware profile. You need a reasonably sophisticated search capability to find the proverbial needle in your haystack(s). Perhaps you are looking for C&C controllers, so you can search network logs for signs. Maybe you are looking for a specific executable loaded, or a particular running process, in which case you would search device logs. Our research shows active testing (as described above) usually provides the quickest way to find infected devices but you cannot afford to overlook logs. Especially to pinpoint potentially dormant malware – which might not yet have executed, or might be inactive in virtual machines, for instance.

2. **Analyze results:** Again, working from your search results, you may need to dig deeper into suspected devices to complete your investigation.

3. **Document results:** As above. Nothing to add here.

4. **If infected, proceed to remediation:** Same as above.

These steps we find all devices on your network that show any signs of the malware you analyzed and profiled. This is not a one-time activity, and we will talk about the need to search on an ongoing basis when we wrap up the process model descriptions.

At the end of this step you know which devices have been compromised. The comes the next big decision what to do with the infected devices.

Remediate

Once we have figured out which devices are compromised, you finally get to address the issues and fix something. But hold your horses — before you start fixing stuff there are decisions to make. You'd think that once you find the malware you'd just clean it up. Right? Surprisingly enough, the answer is *maybe*. A lot more thinking goes into remediation than just a do-or-don't decision. But before we get there we will outline the steps:

> **Remediate:** Finally you need to figure out whether you are going to remediate the malware, and if so how. Can your endpoint agent clean it? Do you need to reimage? Obviously the cost of cleanup which must be weighed against the likelihood of reinfection.

1. **Determine remediation strategy:** Figure out whether you want to clean up the malware, and if so how. Yes, there are situations where you would choose not to clean an infection, as we will describe below.

2. **Remediate:** Once you have decided to clean the device, do it. This may involve removing the malware or wiping the machine entirely, depending on the malware's nature, what it does, and the value of the data on the infected machine. It is generally better to wipe and reimage machines where possible. With modern malware, you cannot be sure you have expunged it using any lesser method, so it's easier and more reliable to just start over. But this research does not judge what is right or wrong for a specific environment. Some organizations try to clean before reimaging, and that is a choice you must make for your organization.

3. **Test remediation:** Regardless of whether you cleaned or reimaged a device, take the time to test your remediation. As with patch management (as described in our Patch Management Quant research), we are talking about software, and software doesn't always work. As the old construction adage goes, "measure twice, cut once." Given the significant ramifications of getting this wrong (the device remains infected), it's important to confirm the fix worked. If so great; if it didn't, try again.

4. **Isolate Patient Zero:** Far too many security folks focus on the initial removal of the malware, but the sad truth is that you are never finished fighting an attack. The old adage about those who forget history being doomed to repeat it, holds for malware as well. Hopefully by finding the devices that were attacked you can understand the malware's trajectory through your environment. If you follow this thought to its logical conclusion you can isolate the first malware victim in your environment, and ultimately identity the initial attack vector that resulted in the compromise. That's what we call Patient Zero. Why is this important? Because you don't want to be infected by the same malware again, so you need to identify and fix the root cause of the attack or be doomed to repeat it.

5. **Inoculate the Healthy:** To continue our healthcare analogy, make sure to protect devices that aren't infected against this specific malware attack. That many involve setting up defenses using other controls (like egress filtering rules on firewalls to block C&C traffic, or host intrusion prevention rules to block registry changes, etc.) to ensure that healthy devices aren't compromised in the future.

To remediate or not to remediate? That is the question.

Earlier we mentioned that you might choose not to remediate a malware attack. We understand that is counterintuitive but it does come up, and you need to consider it. If you are the victim of a targeted attack by a persistent (most likely state-sponsored) attacker – and no, we aren't going to use the acronym – then you may want to quarantine the compromised device rather than simply fixing it.

A persistent attacker *will* have a presence in your environment. That's their mission and they will do whatever it takes, for however long it takes, to achieve and maintain that presence. Once you remediate a compromised device they will initiate another process to gain a new foothold. So the race just starts over. Alternatively you could choose to quietly pull any sensitive data off that machine and then monitor it very closely – perhaps even implementing a special semi-quarantined network where it isn't completely cut off but cannot do much damage. Then you can feed the device disinformation, monitor it, and track the tactics of your adversary rather than tipping them off to compromise a new target.

More likely you are not particularly targeted for this kind of attack; if so then simply carry on. Remediate the device and be sure to keep an eye on things on an ongoing basis. Which brings us to the last step in this subprocess.

Monitor for Reinfection

Let's talk a bit about recursion because detecting malware is not a one-time effort. Malware writers constantly morph and evolve their attacks – using the same base techniques but tuning to account for current detection methods and anti-virus signatures, and to generally improve their wares. That's why we advocate revisiting malware profiles periodically as described in the Malware Profile step.

> **Monitor for Reinfection:** First understand how the malware changes so you can keep your profile current. Then you need to constantly check your infrastructure for signs of reinfection.

But that's not the only recursion necessary in the Malware Analysis process. You need to be constantly vigilant to reinfection by the same attack. Why? Because users continue to click on things. They fall for social engineering attacks and tend to get compromised even when you tell them not to. So you need to keep searching for indications of each attack on an ongoing basis. How frequently depends on many factors, including resources and automation, but need to be clear on the ongoing costs of tracking malware proliferation.

The steps in monitoring for reinfection are:

1. **Define testing frequency:** First figure out how often you want to check whether you have been infected again. This varies depending on the frequency of attack, automation, resource realities, and a general assessment of risk. You can also test some devices – perhaps those with higher risk factors, such as mobile devices and/or those with very sensitive information – more frequently. Clearly you should test as frequently as practical, but every security decision requires a risk/benefit analysis.

2. **Run testing tools:** You defined the rules already and even ran them once when you looked for infected devices, so make the rules persistent on your scanners and other tools and then rerun the tests. You don't need to run all tests all the time but can rely on frequency decisions made in the first step.

3. **Search logs:** As with testing tools, a key to finding infected devices is monitoring the logs of target devices and other network/security devices for indications of attack. This could be an ongoing script run against aggregated logs or a more sophisticated rule deployed on a SIEM. We are fans of continuous monitoring, so if a SIEM is in place we prefer continuous monitoring.

4. **Analyze results:** As if you didn't have enough to do, you need to wade through the findings at some point. Alerting can be your friend here, highlighting specific situations and kicking your incident response process into gear.

5. **Document results:** As in the original process step you need to document your findings — which might be as simple as adding a device name or IP address to the back of a napkin. But if you have many hands in the process, with separate groups responsible for response and remediation, your documentation needs to be a bit more formal. Don't assume the operations team (or whoever is responsible for remediation) has any background on this type of malware; don't assume anything about the impact of the attack; don't assume everybody feels the full urgency of fixing it; don't assume anything. Everything must be spelled out in your documentation.

6. **If infected, proceed to remediation:** If there is a clear sign of infection then continue to remediate, leveraging your top-notch documentation. Obviously this entails many decisions, but we won't put the cart before the horse. We will get into that once we have finished looking for all the indicators of an attack.

Coming Soon: Phase 2

And with that we have described all the steps in the Malware Analysis process model. But we aren't done (not by a long shot), so let's talk about the rest of this research project. Just developing the detailed process map isn't overly helpful to doing your job. So we will take the process model and flesh it out in terms of costs in the next phase of our research. There are three main steps:

1. **Metrics:** For each of the process steps we will determine the most appropriate metrics to track each activity. Most often, this breaks down to the time spent on a certain function. But the function may involve capital or third party service expenditures and we include those as well.

2. **Survey:** As described above, we are pragmatic about how many of these steps real organizations take on a daily basis — not many. But we like to understand what organizations are doing and what they aren't, so we put together a survey tool (and even give away an iPad as an incentive to fill it out) to figure out who does what. No, these survey usually aren't statistically reliable, but they provide real perspective on what folks do and why.

3. **Cost Model:** Finally we pull everything together into a set of spreadsheets you can use to track your activity relating to Malware Analysis to figure out what it *really* costs to deal with all these infections. Yes, it's granular, but without an idea of true cost, it's hard to decide what makes sense to do and what doesn't.

You can participate in our ongoing research by following our blog, where we post everything before it's packaged up as the 'final' document. We appreciate folks who take the time to comment on our posts, tell us where we are a bit misinformed and what makes sense, and share war stories relevant to the research.

If you have any questions on this subject, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com ask us a question via the Securosis Nexus (http://nexus.securosis.com).

About the Analyst

Mike Rothman, Analyst/President

Mike's bold perspectives and irreverent style are invaluable as companies determine effective strategies to grapple with the dynamic security threatscape. Mike specializes in the sexy aspects of security — such as protecting networks and endpoints, security management, and compliance. Mike is one of the most sought-after speakers and commentators in the security business, and brings a deep background in information security. After 20 years in and around security, he's one of the guys who "knows where the bodies are buried" in the space.

Starting his career as a programmer and networking consultant, Mike joined META Group in 1993 and spearheaded META's initial foray into information security research. Mike left META in 1998 to found SHYM Technology, a pioneer in the PKI software market, and then held executive roles at CipherTrust and TruSecure. After getting fed up with vendor life, Mike started Security Incite in 2006 to provide a voice of reason in an over-hyped yet underwhelming security industry. After taking a short detour as Senior VP, Strategy at elQnetworks to chase shiny objects in security and compliance management, Mike joined Securosis with a rejuvenated cynicism about the state of security and what it takes to survive as a security professional.

Mike published *The Pragmatic CSO* <http://www.pragmaticcso.com/> in 2007 to introduce technically oriented security professionals to the nuances of what is required to be a senior security professional. He also possesses a very expensive engineering degree in Operations Research and Industrial Engineering from Cornell University. His folks are overjoyed that he uses literally zero percent of his education on a daily basis. He can be reached at mrothman (at) securosis (dot) com.

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

Our services include:

- **The Securosis Nexus**: The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics that tells you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at https://nexus.securosis.com.

- **Primary research publishing**: We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform to our Totally Transparent Research policy.

- **Research products and strategic advisory services for end users**: Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.

- **Retainer services for vendors**: Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. Even with paid clients, we maintain our strict objectivity and confidentiality requirements. More information on our retainer services (PDF) is available.

- **External speaking and editorial**: Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.

- **Other expert services**: Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: <http://securosis.com/>.