Measuring and Optimizing Malware Analysis: An Open Model

Findings from the Malware Analysis Quant Research Project

Version 1.4
Released: April 2, 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

Executive Summary

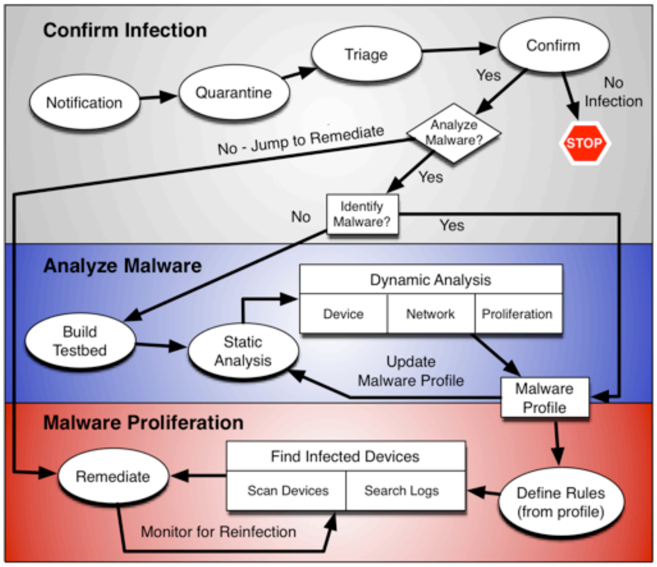
Developing an Open Malware Analysis Model

This report includes the findings of Securosis' Malware Analysis Quant research project. We designed Malware Analysis Quant to kick-start development of a refined and unbiased metrics model for confirming infection from malicious software, analyzing the malware, and then detecting and identifying proliferation within an organization. Our goal is to provide organizations with a tool to better understand the costs of finding and analyzing malware in their operational environments, and to guide improvements through an operational efficiency model capable of capturing accurate and precise performance metrics. Malware Analysis Quant was developed through independent research, community involvement, and an open industry survey of 37 end users in organizations ranging from 75 to 200,000+ employees.

Malware Analysis Process

The process is broken up into 11 separate phases across three subprocesses:

1. **Confirm Infection:** This process typically starts when the help desk gets a call. How can they confirm a device has been infected?
2. **Analyze Malware:** At this point you know there is an infection but not what it is. The next step is to analyze the malware to figure out what it does and how, and communicate that information through a detailed profile.
3. **Assess Malware Proliferation:** Thanks to the last subprocess you know what the malware does, so now you need to figure out whether it's spreading, how much, and what to do about it.



For more detail on the three subprocesses refer to the high level descriptions and the detailed process explanations for Confirm Infection, Analyze Malware, and Malware Proliferation.

Key Findings

1. There is no public, industry-standard process framework for analyzing malware. As a result, Malware Analysis Quant provides a superset framework to encompass activities required across three separate subprocess: Confirm Infection, Analyze Malware, and Assess Malware Proliferation — within any organization, regardless of size or vertical.
2. Organizations in our survey protect the vast majority of their endpoint devices with anti-virus (71% have more than 75% of their devices protected). This indicates the maturity and inertia of traditional anti-virus, and reflects the number of compliance regulations mandating use of AV.
3. Not surprisingly, less than half of our respondents undertake any type of malware analysis. Many, in fact, jump right to remediating an infection most of the time. Only 10% of the 37 respondents always analyze a malware file for indicators of compromise. Many of the respondents do look for the infection elsewhere in their environment, but only a third revisit their malware profiles, so they miss any changes attacks as they evolve, wasting much of the benefit of their profiling effort.
4. In terms of the tools used to analyze malware infections, many of the respondents have forensics and endpoint analysis tools, but far fewer have implemented any kind of testbed to actually study malware and understand its impact. Given the typical focus on containing immediate risk this makes sense, although it is at least very difficult to control malware (or at least reduce its impact) without systematically understanding what it does and specifically looking for proliferation.
5. We understand that the limited response to the survey does not provide a statistically reliable set of data points, but these findings are consistent with our qualitative research with organizations large and small. It seems everyone uses anti-malware technology, to limited effect, and they do much of their clean-up using *ad hoc* techniques without a structured program to analyze attacks and guard against reinfection.

How to Use Malware Analysis Quant

The value of any research is in how you use it to improve your operations and day to day activities. In this paper you will find a very detailed set of process steps, each of which may or may not be relevant to the malware analysis activities within your organization. **Use what makes sense and forget the rest.** In terms of the metrics, an excellent comment on one of our earlier Quant projects puts this initiative into context.

> *Who is the intended audience for these metrics? [Metrics] are part of the job, but I'm not sure what the value is. To me the metrics that are critical around process [focus on whether] the number of changes align with the number of authorized requests. Do the configurations adhere to current policy requirements, etc...*
>
> *Just thinking about [my last] presentation to the CIO, I spent 3 hours getting consensus and 2 hours on prioritizing. [How do these metrics] get me much traction?*

One of the pillars of our philosophy on metrics is that there are really three sets of metrics that any security team needs to worry about. This comment is about the first type: the stuff you need to substantiate what you are doing for audit purposes. Those are key issues, and things you must be able to prove.

The second bucket is numbers which are important to senior management. These tend to focus on incidents and spending. Basically how many incidents happen, how that is trending, and how long it takes to deal with each one. On the spending side, senior folks want to know about percentage of expenditure relative to total IT spending, relative to total revenues, and how that compares to peers.

Then there is the third bucket: *the operational metrics we use to improve and streamline our processes*. This is the crux of the old saw about how you can't manage what you don't measure — the metrics defined in Malware Analysis Quant represent pretty much everything we can measure. That doesn't mean you *should* measure everything, but this project decomposes the processes as far as possible, to provide a basis for useful measurement. Again, not all companies do all the process steps. Actually most companies don't do much from a process standpoint — besides fight fires all day.

Those companies which do many of these processes may not gather quantitative data from those operations, because gathering this kind of data requires a significant amount of effort and long-term commitment to using the data to improve operational activities. If you are trying to understand operationally how much time you spend on things and then use that data to trend and improve operations, or if you want to use metrics to determine whether it even makes sense for you to perform these functions rather than outsource them, you need the data.

Clearly the CIO and other C-level folks are unlikely to be overly interested in the amount of time it takes you to do dynamic analysis on a malware file. They care about outcomes, and most of the time you spend with them needs to be focused on getting buy-in and updating status on commitments you have already made. Which is as it should be.

But if you don't measure and tune your internal processes, odds are you'll be less efficient — eating up budget and being forced to rely on FUD (fear, uncertainty, and doubt) to justify future spending. Which is most definitely how it *shouldn't* be. These metrics provide the fundamental tools for you to optimize your processes, even if you only use a fraction of them.

Table of Contents

Introduction 7

The Malware Analysis Process 8

Confirm Infection Subprocess 9

Analyze Malware Subprocess 9

Malware Proliferation Subprocess 10

Malware Analysis Processes in the Real World 11

Project Assumptions 13

Malware Analysis Quant Metrics 14

Confirm Infection 16

Notification 16

Quarantine 19

Triage 20

Confirm 22

Analyze Malware 25

Build Testbed 26

Static Analysis 30

Dynamic Analysis 32

The Malware Profile 36

Malware Proliferation 39

Define Rules 39

Find Infected Devices 42

Remediate 44

Monitor for Reinfection 46

Conclusion 50

About the Analyst 52

About Securosis 53

Introduction

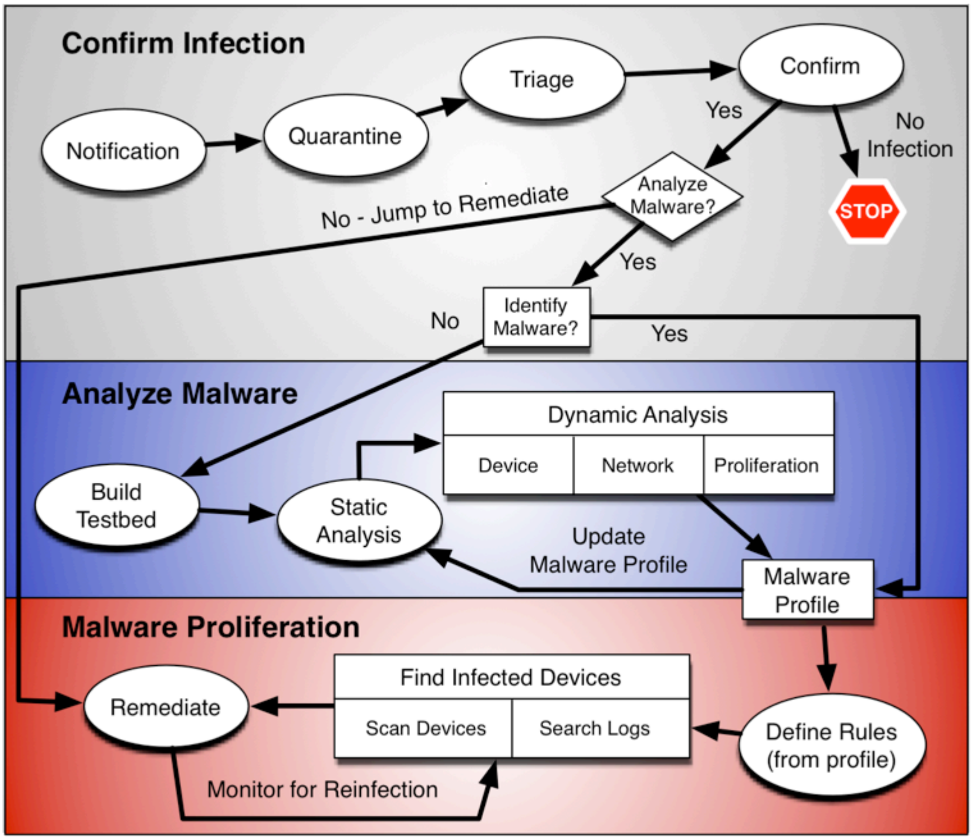
It has been clear for a while that today's anti-malware defenses basically don't work, and as a result way too much malware makes it through your defenses. When you get an infection you start a process to figure out what happened. First you figure out what the attack is, how it works, how to stop it (or work around it), and how far it has spread within your organization. That's all before you can even think about fixing it. To the best of our knowledge, no one has built a specific process map for what this looks like, or a model for figuring out how much it costs to deal with malware on an operational basis.

Clearly the cost is bigger than a breadbox, which is the level of precision most security folks use to quantify their activities. This research is an attempt to understand what it really costs to deal with this epidemic of malware. At least for those who really try to figure out what it does, how it got in, and how to ensure it doesn't get in again to infect other devices. We are pragmatic analysts, and we understand the Quant processes (as described) represent a theoretical ideal and many organizations (especially smaller ones) necessarily take shortcuts or skip steps entirely because of time and/or resource constraints. They clean up the attack using their endpoint suite, or maybe even re-image the machine, and then wait for the next attack. Lather, rinse, repeat.

More mature security operations usually have someone on the team who actually researches the malware and develops a profile of the attack, so the clean-up can be much more effective. But the objective of Quant is not to push a heavy, resource-intensive process onto a group of already overworked security folks. It's to inform and educate practitioners, so if they decide not to do something it is a conscious choice – not an oversight borne out of ignorance. We also see a number of tools to help automate this malware analysis process, and practitioners need to understand how well these tools facilitate their work. This research provides a basis for assessing the cost of doing these tasks manually, and enables you to compare the costs and benefits of automation.

The Malware Analysis Process

We start each Quant project with a detailed process map. Then we describe the processes in gory detail, and define metrics and cost factors for each step. Here is a high level process map for analyzing malware.



We define "Malware Analysis" to encompass how organizations confirm, analyze, and then address malware infections. This is important because anti-malware defenses are clearly insufficient, and far too much malware makes it through. When you get infected, you initiate 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 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 net 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 to 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).

But it doesn't make much sense to totally skip malware analysis and just remediate the device and move on. At minimum you should be looking for other infected devices in your environment based on what you learned during this confirmation subprocess. But to present a complete process map we need to account for the fact that some organizations don't do any type of analysis.

Analyze Malware Subprocess

By now you know there is an infection but probably not what it is. Is it just an annoyance, or is it stealing key data and posing 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 is mostly 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 observe the impact of the malware on the specific device, dynamically analyzing the program to figure out what it actually does. Here you are seeking insight into memory usage, configuration, persistence, new executables, and anything else

interesting associated with execution of the malware. This is done by running the malware in a sandbox. 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. Finally you 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 analysis, which we package up into a malware profile.

With a profile in our hot little hands, we need to figure out how widely it spread.

Malware Proliferation Subprocess

Now that you know what the malware does, you need to figure out whether it's spreading, and if so how widely. This entails 4 more steps:

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 scanning, IDS/IPS rules, asset management queries, etc.

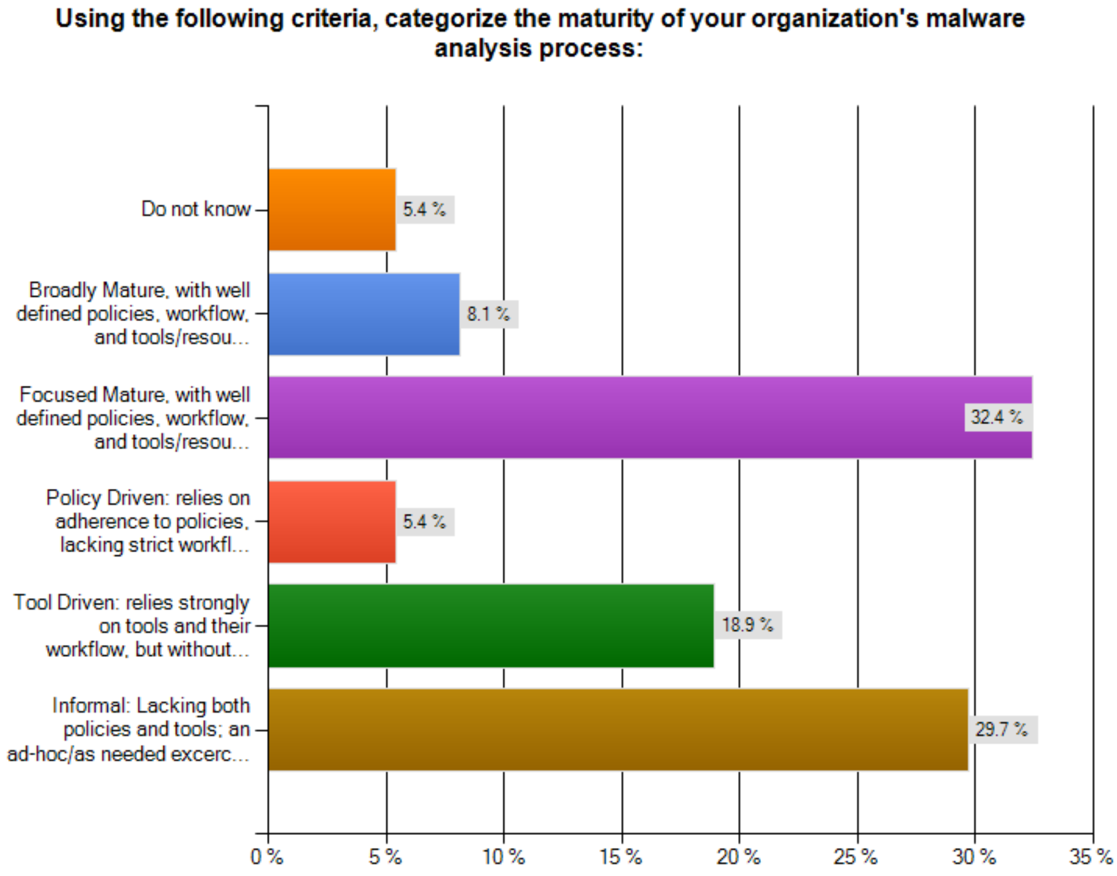
2. **Find Infected Devices:** Then take your rules and use them to search for badness in your environment. This typically entails two separate functions: first run a vulnerability and/or configuration scan on all devices, then search logs for indicators defined in the Malware Profile. If you find matching files or configuration settings, you get alerted to another compromised device. Then search the logs, as malware may be able to hide itself from a traditional vulnerability scan but not to hide its presence from log files. Of course this assumes you are actually externalizing device logs. Likewise you might 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 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 both making sure you update the malware profile as needed and checking continuously for new infections are key parts of the process.

Malware Analysis Processes in the Real World

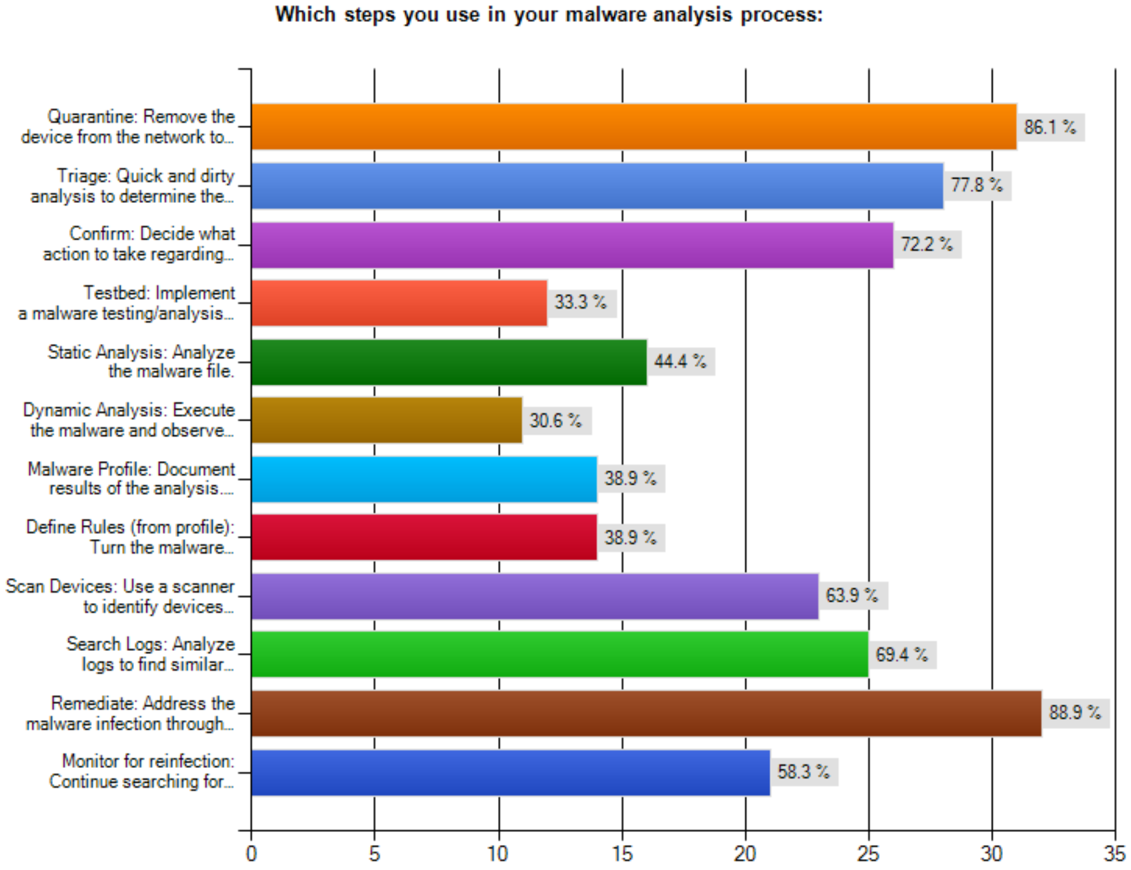
Along with our primary research, we surveyed 37 end users (in organizations ranging from 75 to 200,000+ employees, with an average of 45,000) having anywhere from 100 to 200,000 managed devices in their environment about what processes they have in place. We did this primarily to validate our qualitative research (and add some charts to this report), but also to get a firm grasp on what organizations do in practice, as opposed to pure process modeling theory.



What does this tell us? That organizations are largely split between mature and immature (informal) in terms of malware analysis. In another question, we ascertained that upwards of 75% of respondents used anti-virus technology on 75-99% of their devices. So it's not surprising that *informal* was the second largest group, meaning these organizations take an *ad hoc* approach, based on the situation. Like many other security organizations, they treat malware outbreaks like any other fire. Extinguish it and move on to the next fire.

It is good to see over 40% of respondents assess their program as either focused or broadly mature. Later in this report we will offer some analysis of what tools these more mature organizations use.

For which steps survey respondents undertake, let’s look at the following chart:



What can we learn from this chart? Basically that our small sample does what they need to, including quarantining infected devices and identifying the issue. Many of these organizations do try to figure out whether an infection is part of a larger outbreak, but given the small number that actually analyze malware, it seems this malware proliferation tracking is unscientific at best.

This is not surprising — setting up a testbed and analyzing a malware file is simply not easy enough. But given the need for some level of precision when trying to gauge proliferation, we expect more organizations to look at automated services to understand what malware does and how to find it, in order to prevent further infection.

Project Assumptions

Our design goals for the project were to:

- Build the model to support usage as an operational efficiency model, to help organizations optimize their malware analysis processes, and to compare costs of different options.
- Produce an open model, using the Totally Transparent Research process.
- Advance the state of IT metrics, particularly operational security metrics.

As you read through this report, it's wise to keep the Quant philosophy in mind: the high-level process framework is intended to cover all the tasks involved. That doesn't mean you need to do everything or even that you should, but we offer an exhaustive list. Individual organizations then pick and choose appropriate steps for their own requirements. To meet our goals we made several assumptions:

- *This should be a quantified metrics model, focused on costs:* All the metrics or variables in the model should be measurable with accuracy and precision. 'Qualified' metrics, such as risk and threat ratings, are not included. This model is designed only to measure the costs of analyzing malware, and to identify operational efficiencies or deficiencies in specific process areas. It relies on quantifiable inputs, rather than assessments or other unquantifiable values based on human judgement.
- *The model should apply to all relevant activities in scope:* The scope includes confirming an infection, analyzing the malicious software, and figuring out how widely the infection has spread. Obviously there are many other types of malware-related activities (including configuration management or vulnerability assessment) which could be included, and most of the operational processes would be consistent.
- *The model should apply to organizations of any size or vertical:* This is not designed only for large organizations in particular vertical markets. Although smaller organizations work with fewer resources and different processes, the model still provides a functional framework.
- *The model thus represents a superset of malware analysis activities:* To achieve the dual goals of covering every activity in scope, and applying to organizations of differing sizes and verticals, the model was designed as a superset of any one organization's activities. *We do not expect users to utilize the entire model, and you are encouraged to adapt it for your own particular needs.* We understand collecting all this data could actually cost more than the actual activity of analyzing malware. When in doubt use common sense, and apply the model as appropriate for your environment.
- *The model should break out costs by process to support optimization:* One reason for the extensive detail on each process is to support identification of specific operational efficiencies and problems. Our goal is to help organizations identify and correct problem areas so this project defines all aspects of each process in gory detail to enable data collection, analyses of process efficiency, and trending.
- *This model cannot measure the costs of* ***not*** *analyzing the malware:* Clearly, the easiest way to reduce your malware analysis operational costs to zero is to do nothing. In this project we are concerned only with measuring the costs when you *do* analyze malware. This is because of our strict focus on quantified metrics: we aren't interested in fuzzy math or hocus pocus. If you can't measure it, it doesn't belong in the Quant model.

Malware Analysis Quant Metrics

For each process we have determined a set of metrics to quantify the cost of performing the activity. We designed these metrics to be as intuitive as possible while capturing sufficient detail. The model collects an inclusive set of potential malware analysis metrics, and again you should choose the set that makes sense in your own environment.

Because the model includes so many metrics, we have color coded them to help you prioritize:

Key	The most important metrics in a given category. Using only key metrics will provide a rough but reasonably accurate overview of costs. These are the most useful metrics for determining costs and operational efficiency, and can be reasonably collected by most organizations.
Valuable	Metrics that are valuable but not critical for determining costs and efficiency. They provide greater accuracy than key metrics alone, but require more effort to collect.
Standard	Detailed metrics help with deep quantification of a process, but are either less important or more difficult to quantify. They might be more difficult to collect or involve complex interdependencies with other metrics.

Using key metrics alone will provide a reasonable picture of malware analysis costs and a basis for improving operational efficiency and program effectiveness. Including valuable metrics, or valuable and standard metrics, will provide greater detail.

How to Use the Metrics

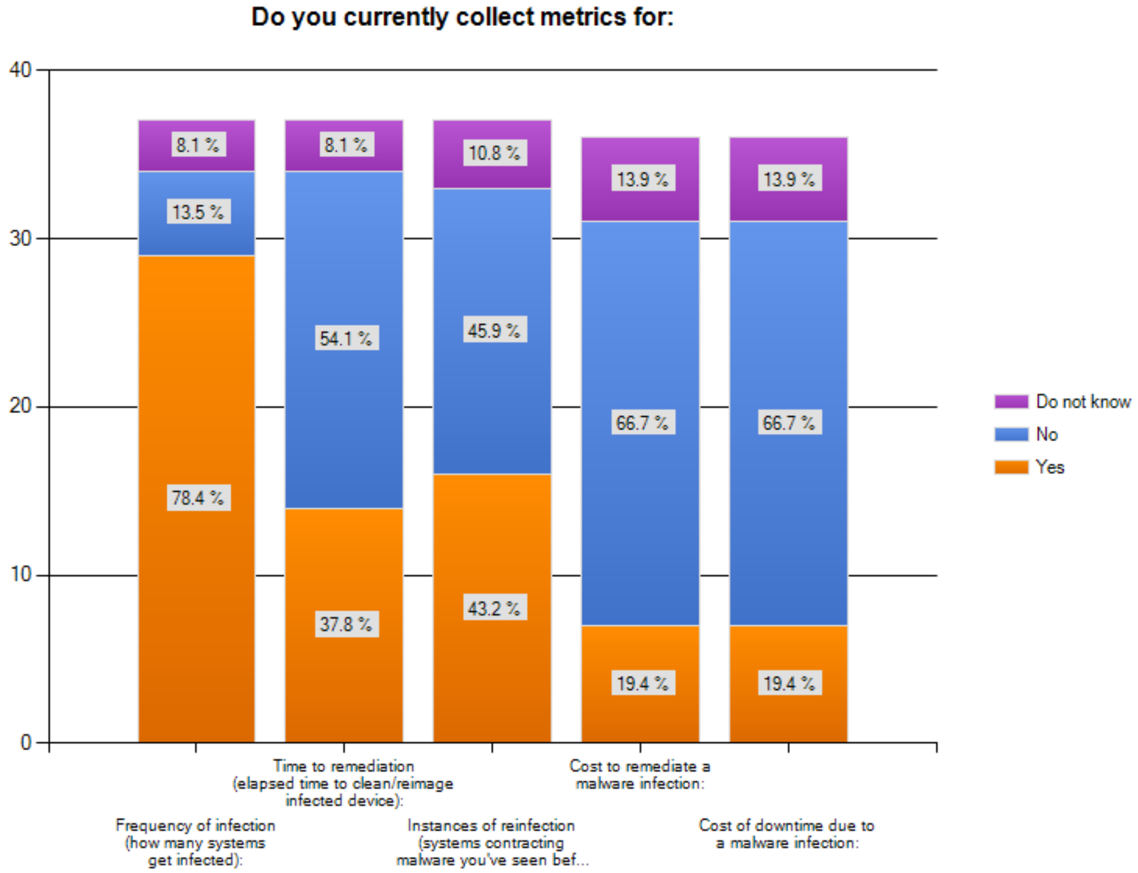
We recommend most organizations start at the process level. That involves matching each process in use within your organization against the processes described in this research before delving into individual metrics. This serves two purposes:

- First it helps document your existing process, or lack thereof. All the metrics in the model correlate with steps in the Malware Analysis Quant processes, so you will need this to quantify your costs.
- Second you may find that these metrics identify clear deficiencies in your current process, even before evaluating any operational changes. This provides an opportunity for a quick win early in the process to build momentum.

We include the applicable metrics for each specific process and subprocess, which can be built up to quantify your entire Malware Analysis program. Make detailed measurements for all the individual processes and then combine them, subtracting out overlapping efforts. Most of the metrics in this model are in terms of staff hours or ongoing Full-Time Equivalents; others are hard costs (e.g., licensing fees, test equipment, etc.). This research project includes a spreadsheet which you can easily adapt to model the malware analysis activities within your organization.

It’s important to keep the purpose of these metrics (and the entire Quant research program) in context. The precision of measurement is less important than consistency and completeness. If you do have the ability to fully quantify costs for each step in the process you’ll get a more accurate result, but that isn’t realistic for most organizations. Still, with the right tools and automation you may be able to come extremely close for certain processes. Metrics vary for any given process, so think in terms of the average cost (or time) for any given step. As long as your method of estimation is consistent you'll have metrics you can work with.

Given the set of metrics and the sophistication of the malware analysis model we have built through this research, let's take a look at what kind of data the survey respondents collect.

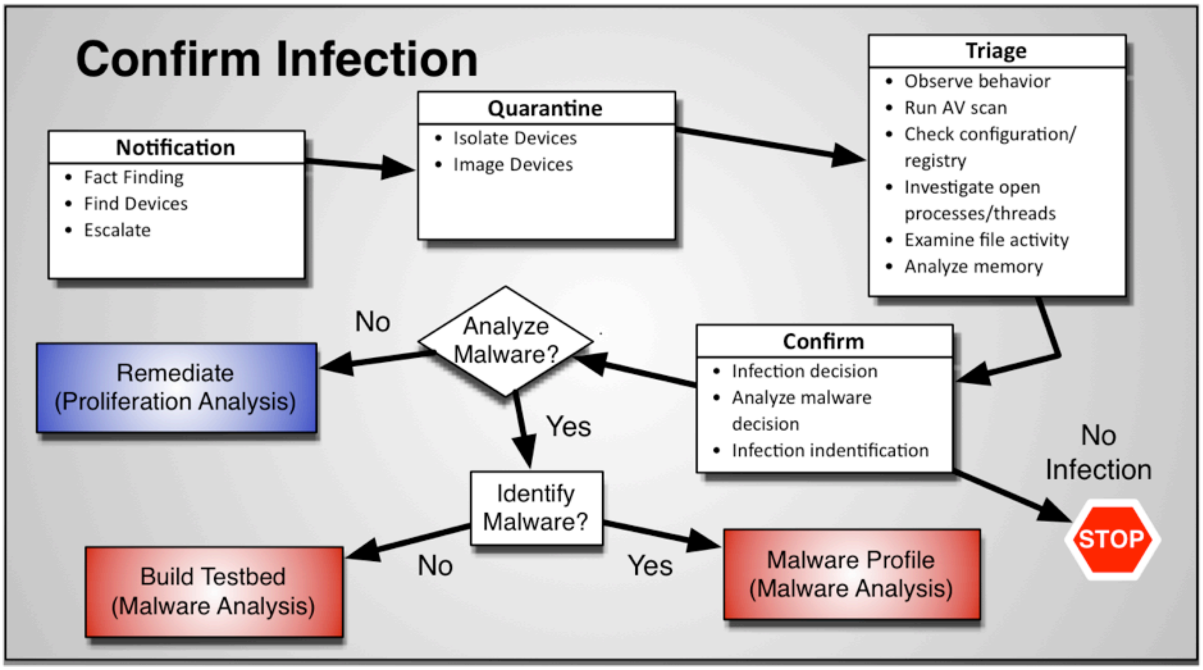


As you can see, a majority of organizations do track how frequently they are infected, with half tracking when they get reinfected. But remarkably few (< 20%) actually map these infections to either a time or cost of remediation. This means many organizations are far from having basic data to take advantage of the Quant model, but tellingly those organizations with a broadly mature or focused mature program do capture a lot of the data to really understand their cost impact from malware.

We took a look at a crosstab of these 15 broadly mature or focused mature respondents and saw the data was a bit more favorable than the broader group. 50% tracked "Time to remediation" and 65% collected data on "Instances of reinfection." That means these self-proclaimed advanced organizations are tracking their infection clean-up over time. That's good news. Although when mapping that to costs, even the advanced groups don't really collect data.

Confirm Infection

You start any malware analysis process by figuring out if you even have a problem by confirming the infection. Obviously until you know there's a problem, there's nothing to analyze.



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 identified it 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 poor? Did a 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 your front-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 might not be, and taking quick action (such as capturing an image for forensics) requires you 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 isolate the machine as quickly as possible.
3. **Escalate:** Now it's time to get the second line of defense involved. These usually aren't help desk operators, but 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 what could be a damaging and rapid infection.

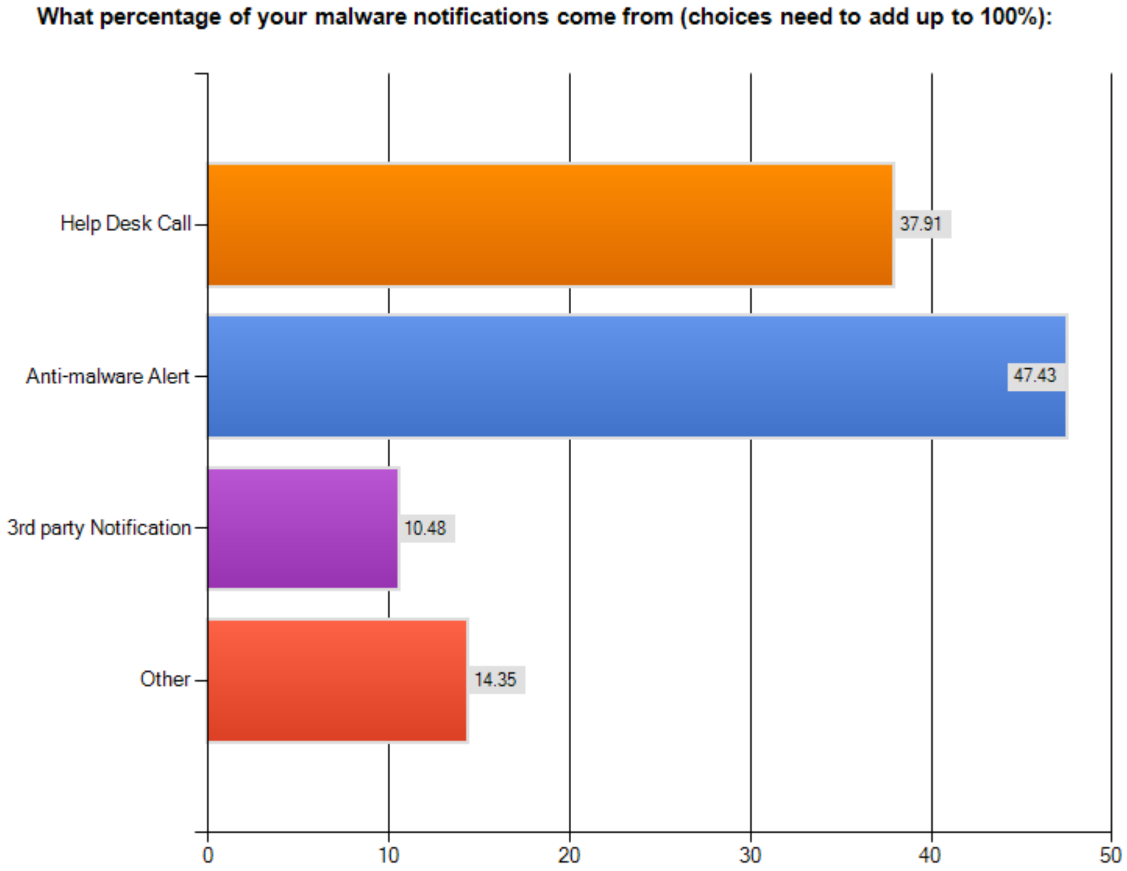
Notification: The process can start in a number of ways, including a help desk call, an alert from a third party (such as a payment processor or law enforcement), or an endpoint suite alert. However it starts you need to figure out whether it's a real issue.

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.

We keep highlighting the need to practice the response process frequently. The bad news is that you are likely to have plenty of opportunities to analyze real malware infections — such is the life of a security professional. But practice is an important part of a comprehensive process model. You don't want to find holes in your response process during a real response.

Notification in the Real World

One of the questions we asked in our survey was how the respondents first learned of an infection. You may hear all of the research from a variety of breach reports, which would lead you to believe organizations typically find out about an outbreak from the FBI. Of course, our limited sample size leads us to a different conclusion, that only a minority of notification happens via a 3rd party. While the majority is from help desk complaints and an alert in an anti-virus console.


What percentage of your malware notifications come from (choices need to add up to 100%):

It would be silly to make any assumptions about the sophistication of attacks based on this kind of question. Anecdotally, our research indicates many of these attacks are less sophisticated in nature (and as such would be detected by anti-malware technology). We believe many of those of advanced attacks aren't detected at all, and that's when a third party notification would kick in — after the damage is done.

Notification Metrics

Variable	Notes
Time to receive notification	
Time for fact finding	Ask questions to validate the issue and identify symptoms.
Time to find device	
Time to determine escalation	The results from fact finding should drive a clear decision process to determine response escalation.
Time to document initial findings	Clear documentation is important for escalation to the next level.

Quarantine

Job #1 of any incident response activity is to contain the damage. 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 such as 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 start poking them yet.

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).

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 users get grumpy when you take their machines, especially if they might lose data. Treat it as a chance to exercise your empathy. But you need to progress to the next step, so the user will need to figure something else out.

Quarantine Metrics

Variable	Notes
Time to decide whether to isolate device	The criteria for when to pull a device from the network should be clearly defined.
Time to isolate device	
Time to capture and store device image	The image is captured both for forensics and investigative purposes.

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. Of course, depending on the nature of the potential infection, you might skip some steps. Or you might do slightly different ones, but the point is to have 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.

Triage: With the device off the net you have a chance to figure out how sick it is. This 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.

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 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:** Also check a device memory dump for problems. Yes, you will need a sophisticated tool for this, but often the only traces of 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 might 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 break down what the tools do in order to provide an accurate idea of the manual 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 determine the next step.

Triage Metrics

Variable	Notes
Time to observe device behavior	Here you are looking for something obvious. Like a zillion pop-up windows.
Time to run AV scan	You need to cover all your bases, and sometimes the AV scan actually finds something.
Time to test configuration/registry	
Time to analyze open processes	
Time to examine file activity	Malware usually changes or otherwise impacts files.
Time to analyze memory	

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.

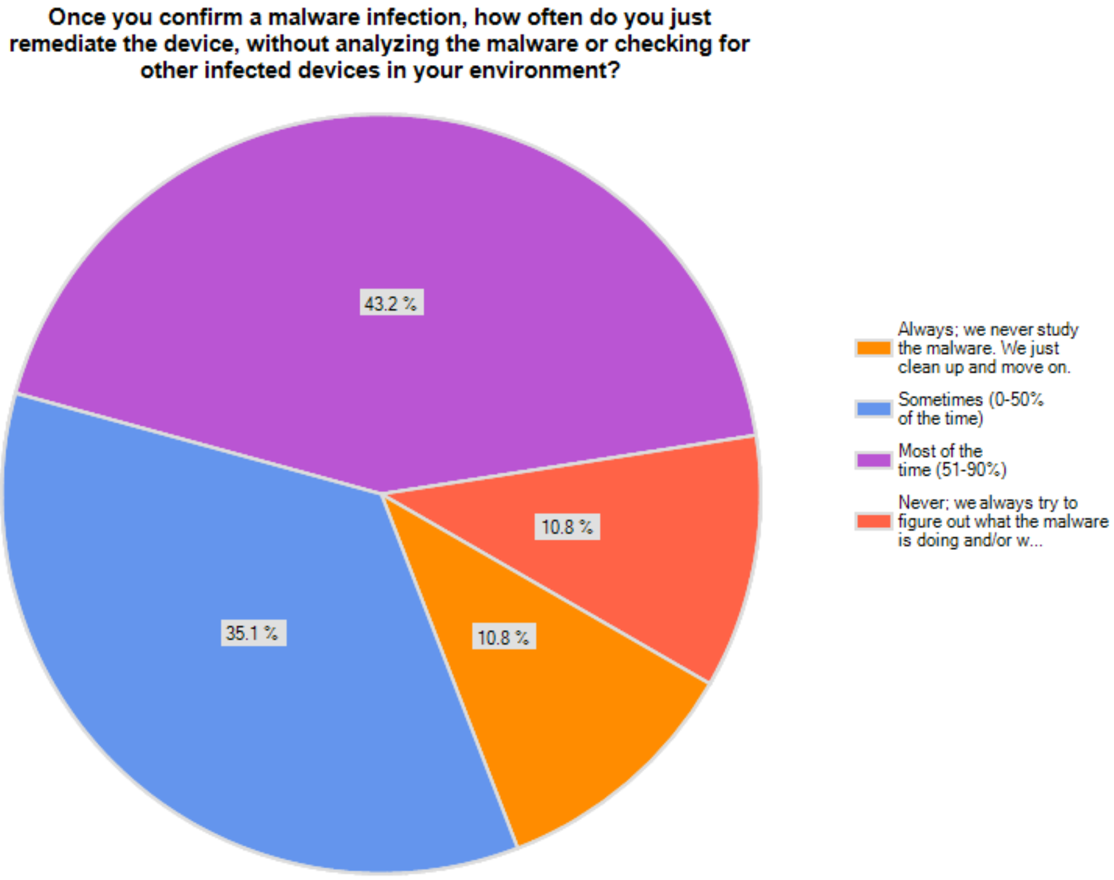
Confirm: At this point you should have enough information to know whether the device is infected and by what. Now you have to decide what to do next.

1. **Infection Decision:** From triage you should know definitively whether the device is infected. If not, you are done – great. If it is 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 a detailed analysis to profile the malware, or a simpler one just to see whether there are other devices infected with similar malware already in your environment. As we mentioned above, we don't recommend just remediating a device and move on, but it happens, so we factored that into the process map.

3. **Infection Identification:** The device is infected and you have decided to do some type of malware analysis. Next you need to figure out whether you know what it 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 other devices on your network. If that's the case skip malware analysis and jump directly to Malware Proliferation activities. If you don't know what the infection is, then your next step is the Analyze Malware activities.

As we mentioned when talking about the maturity of many organizations' malware analysis processes, it's not surprising to see many organizations jump directly to the Remediate step (as opposed to analyzing much of anything). The survey shows that many organizations just fix the immediate issue and move onto the next one.

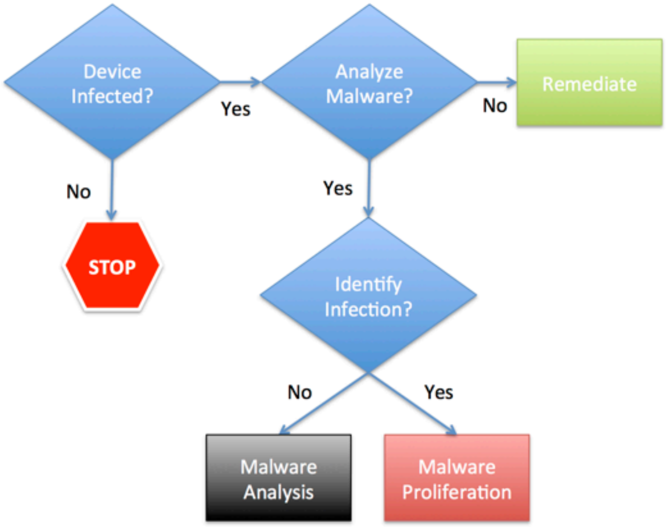


These answers indicates that malware analysis is regarded as a luxury, and only about half (10% always analyze and 43% analyze most of the time) consistently analyze how they are being attacked. As the process maps show, this is a complicated process. We don't have data to substantiate this, but in other markets we have seen increased uptake on analysis functions when the process gets easier and cheaper via automation. We expect a similar uptake in malware analysis activity as the capabilities mature, but again that is based on parallel experience rather than data.

Confirm Metrics

Variable	Notes
Time to decide whether the device is infected	Based on the triage results, using criteria defined in advance for determining whether a device is infected.
Time to attempt to identify malware	Check internal and external resources to attempt to identify the infection.
Time to document findings and decision	Regardless of the decision in the next step, you need to document what you've found.
Time to escalate/hand off to next step	Depending on the decision, the file and documentation either goes to one group to analyze malware, or to another group to assess proliferation.

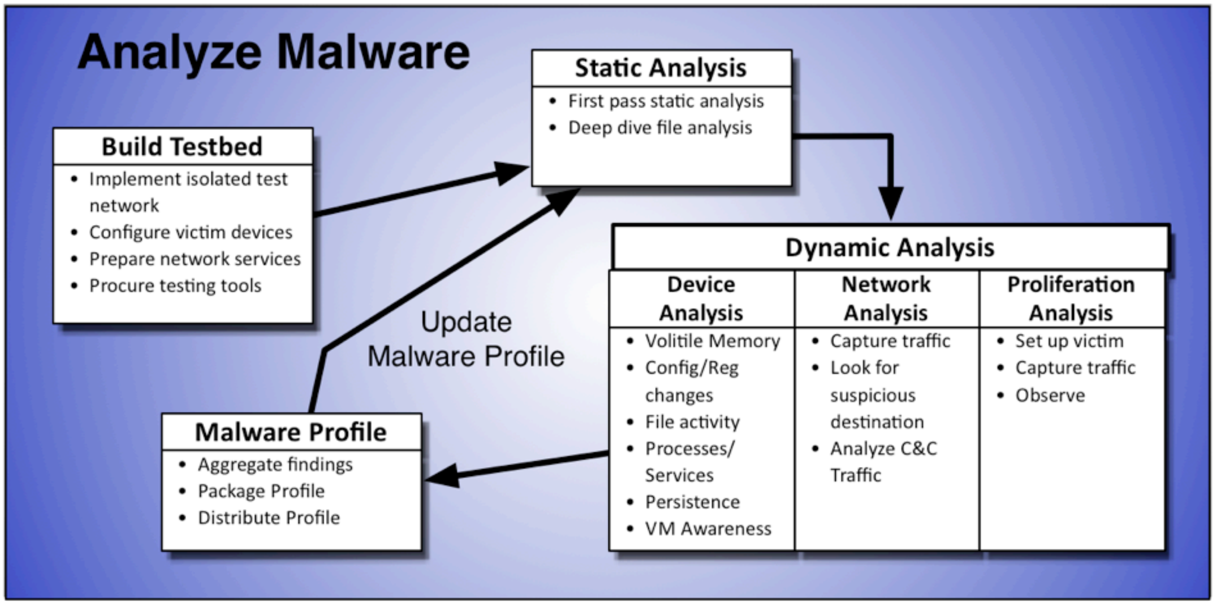
So your decision tree looks like this:



Based on this tree, either go directly to the Remediate step, proceed to the Analyze Malware activities, or start figuring 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 posing a clear and present danger to the organization? The next subprocess digs into malware files using a variety of 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 not every company does all these steps. In fact most organizations don't perform many of these functions — instead they rely on 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 Quant is to educate readers about all the things that can be done, and provides enough information 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 mostly a one-time operation but keeping your environment and tools current requires ongoing effort. For now let's just say a bit about the environment you should strive to build.

Of course we need to mention caveats. Some of you will not need all 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.

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 is mostly a one-time effort, but you will always be adding to the testbed based on the evolution of your attack surface.

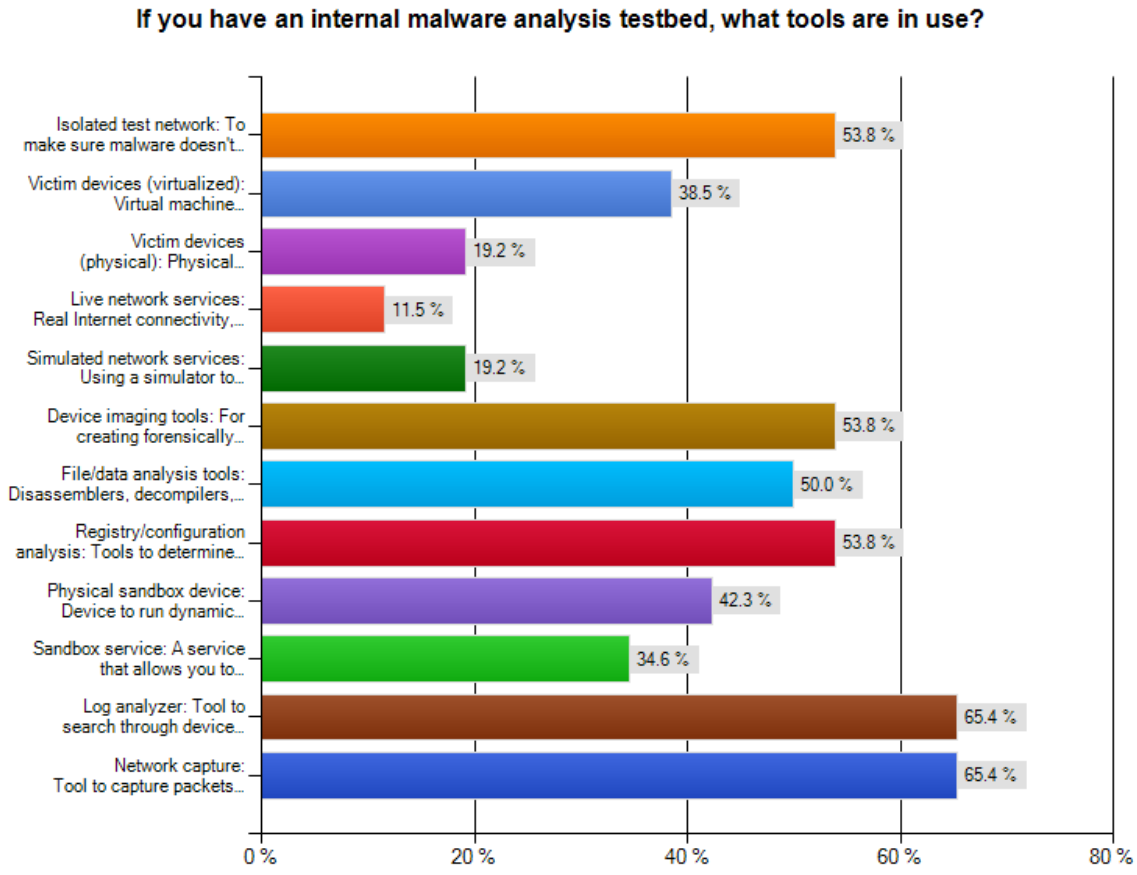
- **Isolated Test Network:** Don't test live malware on production networks — you need a separate network for your testbed.
- **Victim Devices:** You need victim machines in a variety of configuration states (patched, unpatched, etc.) and a variety of operating systems from your environment. You will 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 need 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 our research shows that 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 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 malware activity, so you need some kind of log aggregator and parser to wade through what could be thousands of log records.
- **Network Capture:** You 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. Once we have the testbed in place we can analyze bad stuff.

Real World Testbeds

For those organizations (as shown above, approximately 33% of the survey respondents) who have a malware analysis testbed, what kind of equipment do they have in place? Here is what the survey says:



Not surprisingly, given that only 50% of respondents analyze malware consistently, we see 30% have a formal malware analysis testbed. But we see the number of respondents who perform log aggregation, and have the ability to do some kind of packet capture, as positive. We also see over 50% penetration of device imaging and file analysis tools, which are instrumental for forensics. And about 54% have an isolated test network.

What these respondents don't have are the tools to perform dynamic analysis of live malware, which is not surprising given the risk of infecting yourself. We expect the 30% of folks using a sandbox service to grow over time, as they allow the dynamic analysis to be performed (by someone else) outside your network. We believe those services will largely take over the dynamic analysis aspect of analyzing malware over time.

Build Testbed Metrics

Variable	Notes
Time to research and design testbed	This should include both the equipment required (network, devices, etc.), as well as external data sources.
Cost of equipment and tools	
Time to build isolated test network	
Time to configure victim devices	You will likely need both virtual and physical victims.
Time to install network services	May include DNS, Internet access, IP and/or file reputation data sources.
Time to research and select testing tools	
Time to install/configure testing tools	May include device imaging, file/data analysis, registry/configuration analysis, sandbox, log analyzers, and/or network capture/analysis tools.
Time to build repositories for comparison	As you analyze you will have data to compare new attacks against, so this step involves building the environment to store and index other malware analysis findings.
Time to determine format/structure of malware profile	The profile is how you will communicate findings about the malware to other operational groups.
Revisit tool selection and testbed design, as needed	You should survey the latest and greatest in analysis tools fairly frequently to take advantage of innovation.

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 your file analysis. The output of this analysis is a profile of the malware attack, used for searching for 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 actual analysis step is 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. Check the file's fingerprint (usually an MD5 hash) to 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 file compression, mostly to save space. But attackers also pack to obscure content and/or behavior, so you need to figure out whether the packing process hides badness — because 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 reported 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 like 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 are ready to perform an actual static analysis.

But of course there is no panacea – figuring out what the malware does is detective work. And 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.

Static Analysis Metrics

Variable	Notes
Time to check file fingerprint	Check the file hash against database(s) of known malware files.
Time to analyze file packer	
Time to classify file structure	
Time to analyze text strings	This involves finding clues in snippets of the malware and searching for similar snippets to find potential patterns.
Time to disassemble malware	Using a disassembler can provide a lot of insight into what the malware is doing, and again can provide identifiable patterns to help identify the attackers.

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. Start by running the malware in your testbed. Each aspect of this analysis involves looking at a 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 begin dynamic analysis. Let's look at the types of information to gather and why.

- **Volatile Memory:** Malware can overflow buffers and/or tamper with program memory to gain access to a device. By capturing and then analyzing the device memory you can figure out how the malware uses memory.
- **Configuration/Registry Changes:** Look for any evidence of the malware changing configurations and registry keys. You expose 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 may also 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 will 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.
- **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 know what processes/services were running when you set up the victim device, so you can pinpoint what the malware changed.

Device Analysis: First observe the impact of the malware on the specific device, dynamically analyzing the program to figure out what it actually does. Here you are seeking insight into memory usage, configuration, persistence, new executables, and anything else interesting associated with execution of the malware. This is done by running the malware in a sandbox.

- **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. Analyzing changes to the configuration, registry, and files should show whether or not, the malware attempts to restart itself, and how.
- **VM Awareness:** A lot could 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 all 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 with all this data.

Network Analysis

As we mentioned earlier, 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 its 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: Once you understand 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 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 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. Controllers often 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 realize 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 to test malware. Isolation need not be simply a different IP address – it could entail a separate Internet Service Provider or even geographic egress point. Don't irritate or panic attackers into taking down your production network.

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 an automatic attempt to spread internally? As usual we start the proliferation analysis by running the malware. Of course you can run the malware once and capture all the data for the device, network, and proliferation analyses concurrently.

Proliferation Analysis: Finally you 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.

- **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 performed 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 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*. You need this analysis to develop a malware profile. There are services that can analyze your files and send back a report of what they find, but Quant models the costs of performing the actual analysis, so we cannot skip this step.

Dynamic Analysis Metrics

Process Step	Variable	Notes
All	Time to run malware against victim devices	Match the file hash against database(s) of known malware files.
Device Analysis	Time to capture and analyze volatile memory	
	Time to analyze configuration & registry changes	
	Time to assess and log file activity	
	Time to capture and analyze processes & services	
	Time to restart victim to test persistence	
	If no visible impact on VM, time to test against a physical machine	Testing VM awareness generally requires a physical victim device, rather than a virtualized victim.
Network Analysis	Time to capture network traffic	
	Time to search for suspicious destinations	Using IP reputation and C&C analysis.
	Time to analyze C&C traffic	Determine what is being sent and where.
	Time to analyze exfiltrated information	If available.
Proliferation Analysis	Time to set up another vulnerable victim	This additional victim will be the target of any attempts by the malware to spread, pivot, or otherwise infect another device.
	Time to capture network traffic (again)	Rather than initial traffic patterns, you are now looking to see how the malware searches for devices and follows up.
	Time to isolate reconnaissance traffic	
	Time to observe and assess proliferation activity	Malware may use different tactics to compromise additional devices once established; so observe not just for the initial attack vector, but for anything else.

The Malware Profile

We wrap up the Malware Analysis subprocess by leveraging all the analysis done in the previous couple steps (Static 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 malware.

> We need to keep our goals in mind when building the profile.
>
> - **Assess Malware Proliferation:** There are times when only one device gets infected during a malware attack. But other attacks become full-blown outbreaks. So your first job after developing the profile is to figure out whether the malware has spread. That is our third subprocess.
> - **Prevent reinfection:** The other purpose of the malware profile is to make sure you don't get reinfected.

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 might pull from several different places. Here is a short list of information types you might have include.
    - File attributes
    - Registry settings
    - Processes/Services
    - New executables
    - Domains/Protocols
    - 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 affect the type and depth of documentation you produce.

We don't normally 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.

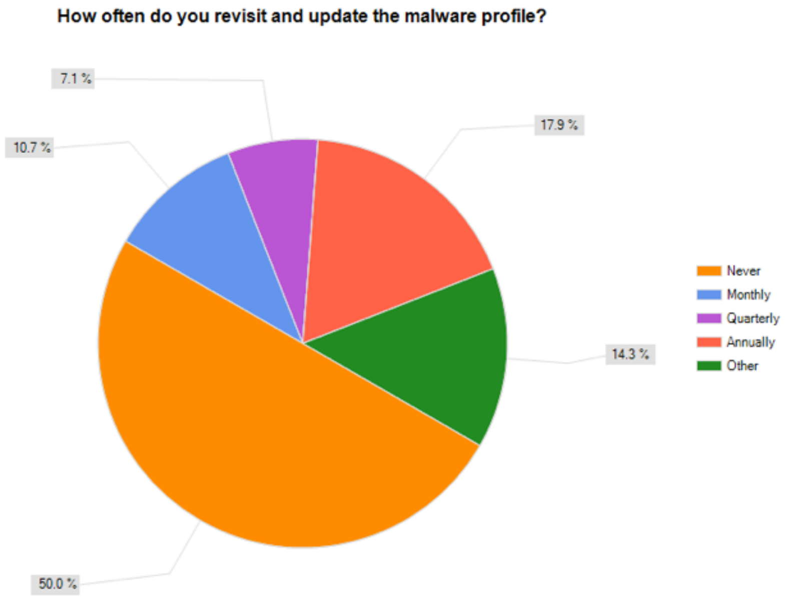
Revisiting the Malware Profile

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 updates to the profile, and reinfection implies a 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), the research community continues to find different variants, which means you need to update your profile.

With a very detailed and specific profile the incident response team can try to figure out whether the malware has spread throughout your organization, and if yes then how badly.

So do folks really take the time to go back and revisit their malware profiles? Based on our responses, not much.



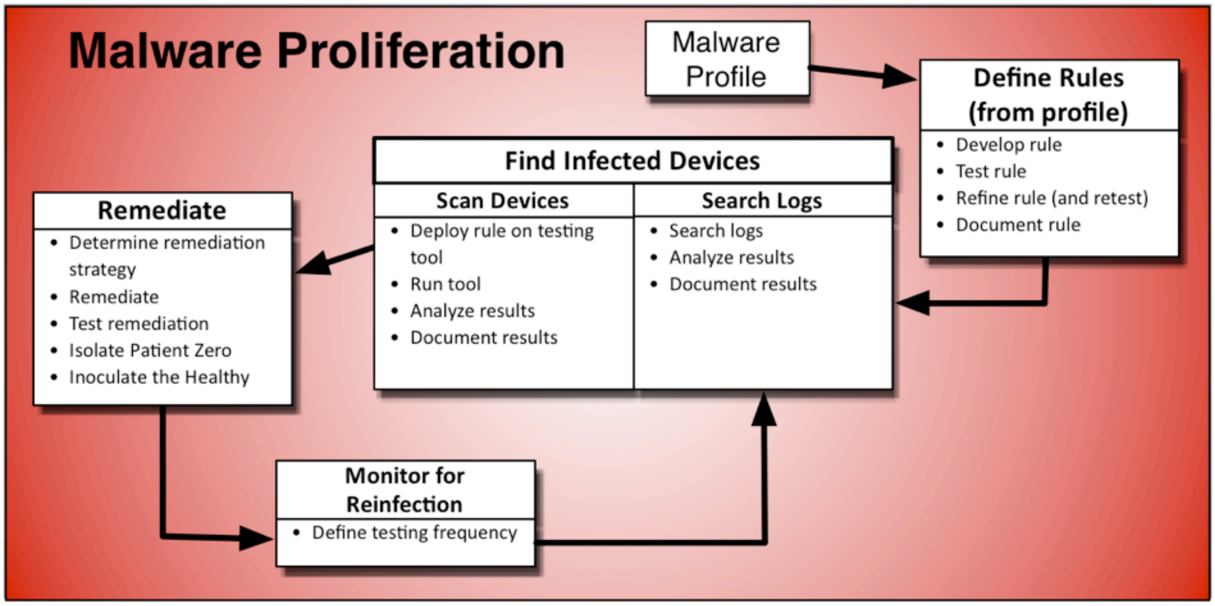
Is this indicative of a huge problem? It's not optimal, but the reality is that with the vast majority of organizations using traditional anti-malware technology, the research teams of the AV vendors do track the variants and new attack vectors of known malware. Again, it's not perfect, but it's better than nothing.

Malware Profile Metrics

Variable	Notes
Time to aggregate findings	Gather the indicators identified during the analysis steps, including file attributes, registry settings, processes & services, new executables, domains & protocols, command and control activity, and persistence.
Time to document findings	You should have a standard format for the profile, depending on the operational constituencies who will use it.
Time to distribute the profile	You need to include the time to deliver the information and perform the formal hand-off to make sure nothing falls through the cracks.
Time to revisit the profile	Malware is not a static entity, so you need to budget time to revisit each malware profile periodically to account for changes in attack vectors, payloads, etc.

Malware Proliferation

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



Define Rules

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

1. **Develop Rule:** First develop the rule. Sorry, but Quant is inherently pedantic. 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 the above.

2. **Test Rule:** Now you get to test your shiny new rule in your testbed. That means actually infecting a victim machine (safely bottled up, 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, fix it and verify, and *then* move on.

3. **Refine Rule (and Retest):** After failing the test (or perhaps just not exactly passing), make the necessary 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 run the scans or implement the rules on devices they control.

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 scanning, IDS/IPS rules, asset management queries, etc.

5. **Return 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 1 and start building the next rule, based on the next indicators in the profile. Remember, the more tightly 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 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 try 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. You will 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.

Define Rules Metrics

Variable	Notes
Time to analyze malware profile	Determine how to search for each indicator defined in the profile. Can you scan for it with a tool? Search logs?
Time to develop rules/queries for the indicator	This can be tedious, but the tighter you make the search criteria, the fewer false positives you will need to deal with later.
Time to test rules/queries	You need to set up a vulnerable device with the malware, on an isolated network, and then make sure your rules/queries actually find it.
Time to refine (and retest) rules/queries	You will likely have to iterate a few times to get a set of rules/queries that work well. Again, the longer you spend getting the rules right, the fewer mistakes you'll need to track down.
Time to document rules/queries	As with all good processes, document what you did, and hopefully why.
Repeat for each indicator in the profile	Lather, rinse, repeat. You'll need a rule and/or query for each indicator identified in the profile.

Find Infected Devices

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

Scan Devices

We start with testing tools.

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 the cost model.

3. **Analyze Results:** Once the tool finishes, 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 actually been compromised.

4. **Document Results:** If you are performing the analysis and scanning, then the documentation could 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 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.

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, to make a bit later.

Search Logs

Searching logs is very similar to running testing tools.

1. **Search Logs:** As with 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) is generally the quickest way to find infected devices, but you cannot afford to overlook logs. Especially to pinpoint 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 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 them.

Find Infected Devices Metrics

Process Step	Variable	Notes
Scan Devices	Time to deploy rule on testing tool	Load the rules developed earlier into the scanner or other tool.
	Time to run rule	
	Time to analyze results	Identify false positives and prioritize which devices have the most serious issues.
	Time to document results	Prepare documentation for the ops teams tasked with remediation.
	Time to escalate infected devices to remediation	
Search Logs	Time to aggregate logs	This can (and should) be leveraged with a log management initiative. It usually entails setting up collection from monitored devices. See Network Security Quant for detail on how.
	Time to run *ad hoc* search queries	Based on the queries defined for the malware, search the aggregated log data to identify potentially compromised devices.
	Time to analyze results	Identify false positives and prioritize which devices have the most serious issues.
	Time to document results	Prepare documentation for the ops teams tasked with remediation.
	Time to escalate infected devices to remediation	

Remediate

Once you have figured out which devices are compromised, you finally get to address the issues and *fix something*. But hold your horses — first you have some 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 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 decide not to clean an infection, described 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, its impact, 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 you might choose differently in light of your own particular constraints. Some organizations try to clean before reimaging, and that is a choice you must make.

3. **Test Remediation:** Regardless of whether you cleaned or reimaged a device, take the time to test your remediation. As with patch management (described in Patch Management Quant), we are talking about software, which doesn't always work. As the old adage goes, "measure twice, cut once". Given the serious 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 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 through, you can isolate the first malware victim in your environment, and 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 face it again, and again.

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 (such as 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 is a valid choice which you need to consider. 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 again. 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 it 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 and be sure to keep an eye on things on an ongoing basis.

Remediate Metrics

Variable	Notes
Time to determine remediation strategy	Do you fix the device? Wipe it? Do something else?
Time to gain consensus on remediation strategy	Make sure everyone agrees, especially if the decision is *not* to remediate for some reason.
Time to remediate device	
Time to test remediation	Today's malware is hard to kill, so you need to make sure you have really gotten rid of it, unless you wiped the device.
Time to isolate "Patient Zero"	Identify initiator/root cause of the infection to ensure all instances are identified and remediated.
Time to determine whether inoculation is necessary	Do you need to change a configuration setting or implement a specific control to address this infection?

Variable	Notes
Time to inoculate, if necessary	Implement the additional controls and/or change the configurations.

Monitor for Reinfection

We have finally reached the last step in this whole process. So let's talk a bit about recursion — detecting malware is not a one-time effort. Malware writers constantly morph and evolve their attacks – using the same basic techniques, but tuning to counter current detection methods and anti-virus signatures, and generally to improve their wares. That's why we advocate revisiting malware profiles periodically, as described in the Malware Profile step.

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? Current malware processes don't address the root cause of the infection, which is typically a dropper or some other means of insertion. So as soon as you clean the device, the dropper brings down the malware again.

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.

You also have to factor in *user reality*. Basically that your 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 you need to be clear on the ongoing costs of tracking malware proliferation.

1. **Define Testing Frequency:** First figure out how often 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 those with particularly 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 them persistent on your scanners and other tools and 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 recommend continuous monitoring.

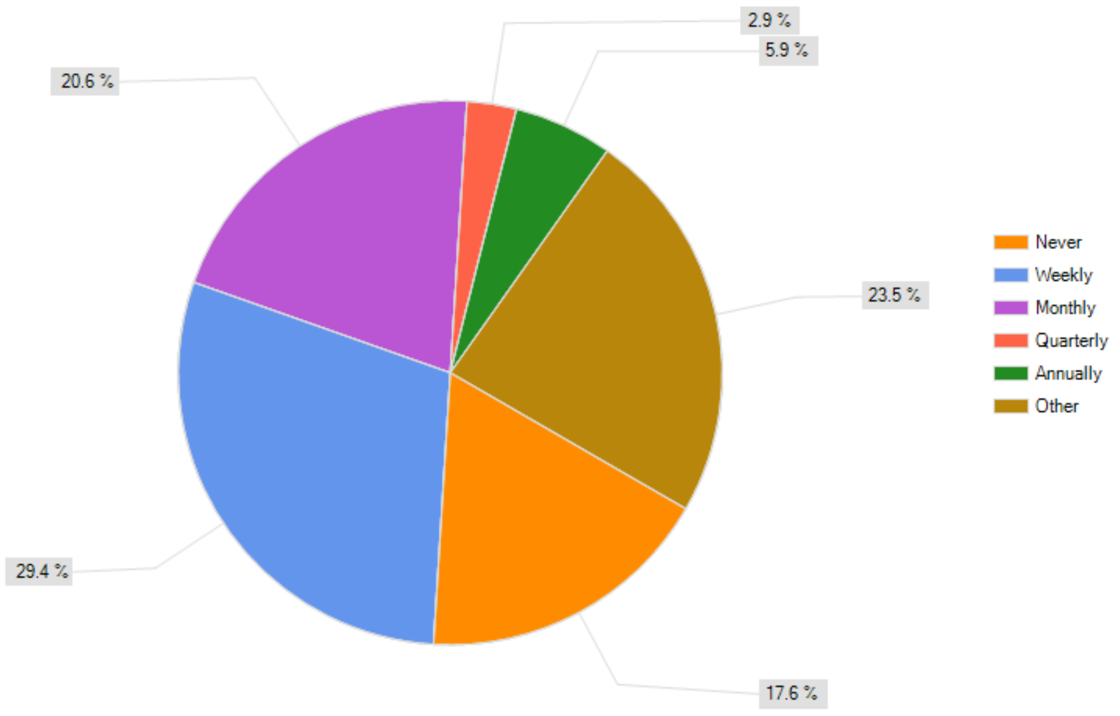
4. **Analyze Results:** As if you didn't have enough to do, you need to wade through the findings at some point. Alerting can be your friend here, highlighting specific situations and kicking your incident response process into gear.

5. **Document Results:** Just like 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 many hands are involved 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, which we will get into once we have finished looking for all the indicators of an attack.

Do folks really take the time to monitor their environment for reinfection? Most of our respondents do monitor their devices on an ongoing basis. Given the pain of having to fix something again, it is unsurprising that respondents check for future outbreaks.

If you monitor your environment for reinfection, how often do you check for the same malware?



Interestingly, in this survey question, we allowed respondents to pick “other” and other tended to mean “continually until eradicated.” Which makes sense. Another noted they start monitoring/scanning hourly, then daily, then monthly as they get further away from the time of infection. Again, this represents a good security practice. Though we do see the same attacks over and over again, so we continue to believe that an ongoing practice of searching for all of the malware you’ve seen gives you the best opportunity to prevent reinfection.

Monitor for Reinfection Metrics

Variable	Notes
Time to run testing tool periodically	More appropriate: Time to load rule into tool, test effectiveness, and build testing schedule – automation and scheduling yield huge dividends.

Variable	Notes
Time to run search queries	More appropriate: Time to load rules into a SIEM or Log Management product, test rules, and set appropriate alerting thresholds. Searching for malware is rarely the sole justification for buying a SIEM, but is one way SIEM justifies its investment.
When receiving an alert, analyze results	Similar to the Find Infected Devices section – you need to figure out whether an alert represents an actual infection.
Document results of alert	Assemble information for other teams to use to remediate the infection.
Go back to remediation if new infections are found.	You found something – now fix it.

Conclusion

Malware continues to plague enterprises large and small, imposing significant costs for clean-up and causing frequent reinfections. But *how much* does it really cost? The Malware Analysis Quant project offers a very granular process map and defines the costs of performing the relevant tasks. Organizations can use it to quantify their costs for analyzing malware and to optimize their processes based on those costs.

We supplemented our primary research with a small survey (37 respondents) to validate our positions. We learned the following (also discussed in the Executive Summary):

1. Organizations in our survey protect the vast majority of their endpoint devices with anti-virus (71% have more than 75% of their devices protected). This likely results from both the maturity and inertia of traditional anti-virus, as well as a number of compliance regulations mandating the use of AV.
2. Not surprisingly (but still discouragingly), less than half our respondents undertake any type of malware analysis. Many, in fact, jump right to remediation, and just fix the infection most of the time. Only 10% always analyze malware for indicators of compromise. Many respondents do look for the infection elsewhere in their environment, but only a third revisit the malware profile to keep it current; given the rapid rate of change in malware, that reduces the effectiveness of their profiling efforts.
3. In terms of the tools used to analyze malware infections, many respondents use forensics and endpoint analysis tools, but far less have implemented any kind of testbed to actually study the malware files and understand their impact. Given the typical focus on containing the immediate risk, this makes sense although it's hard to see how the we can control malware (or at least reduce its impact) without a systematic understanding of the malware and its behavior, and without specifically looking for it within the environment.

Although the survey shows where the industry is lagging in terms of systemic and consistent malware analysis, we expect organizations to increasingly embrace malware analysis. Adoption will be driven by growing understanding that without analyzing the attacks and assessing proliferation, costs to remediate the same attacks over and over again will skyrocket. Accordingly, we anticipate sandboxing services becoming more integral to the process over time, as these offerings allow organizations to benefit from sophisticated dynamic malware analysis without having to invest in a testbed or risk running live malware in their environments. Fortunately the information from a sandbox service can be factored directly into remediation and ongoing monitoring efforts.

As with most emerging technologies, mature organizations take a leadership role early in the adoption cycle. As the technology matures, both in terms of ease of use and cost, more companies adopt the technology until it becomes critical enough to integrate into existing platforms. We expect malware analysis to become an integral part of existing endpoint protection platforms, but not immediately.

Where to Start?

We offer many operational steps and associated metrics in this project. We recommend organizations start small. Attack one aspect of the process model at a time, achieve some early success, and then gradually expand the scope. It probably makes the most sense to look at one of the more mature process models — perhaps confirming the infection or analyzing proliferation. It can be anything, so long as the number of participants is reasonable and the amount of data gathered is manageable.

The steps to introduce this Quant approach to your organization are pretty straightforward and very replicable.

1. Pick a place to start.
2. Map the process.
3. Choose the metrics.
4. Collect the data.
5. Analyze the data.
6. Adapt the process.

Then go back to Step 1, with another subset of your malware analysis operational processes. We don't mean to oversimplify but it's not hard. Your organization just needs the commitment to systematically collect data and adapt the processes based on what the data tells you.

Finally, the authors of this report would like to encourage additional open, independent, community research and analysis projects in IT and security metrics. A transparent research process enables new kinds of collaboration, capable of producing unbiased results. We are investigating other opportunities to promote open research and analysis, particularly in the areas of metrics, frameworks, and benchmarks.

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 or ask us 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 eIQnetworks 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 to tell 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 introduce 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 evaluation, and risk assessment.

- **Retainer services for vendors**: Although we 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 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 each 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/>.