Evolving Endpoint Malware Detection: Dealing with Advanced and Targeted Attacks

Version 1.3
Released: July 12, 2012

Securosis, L.L.C. 515 E. Carefree Highway 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 Trusteer

Trusteer is the leading provider of cybercrime prevention solutions that protect organizations against financial fraud and data breaches. Hundreds of organizations and millions of end users rely on Trusteer to protect their computers and mobile devices from online threats that are invisible to legacy security solutions. Trusteer's Cybercrime Prevention Architecture combines multi-layer security software and real-time threat intelligence to defeat zero-day malware and phishing attacks, and help organizations meet regulatory compliance requirements. Leading organizations such as HSBC, Santander, The Royal Bank of Scotland, SunTrust and Fifth Third are among Trusteer's clients.

For more information visit: www.trusteer.com.

Copyright

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

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

Table of Contents

Control Lost 4

Behavioral Indicators 7

Providing Context 11

Controls 14

Trade-offs and Compromises 18

Summary 21

About the Analyst 22

About Securosis 23

Control Lost

With malware front and center in the minds of all security professionals, we have focused a lot of our research on the evolution of malware and emerging controls to deal with it. We started a few years back by documenting Endpoint Security Fundamentals, and more recently looked at network-based approaches to detect malware at the perimeter. Finally we undertook the Herculean task of decomposing the processes involved in confirming an infection, analyzing the malware, and tracking its proliferation with Malware Analysis Quant.

Since you were but a wee security analyst, the importance of layered defense has been drummed into your head. No one control is sufficient. Stacking as many complimentary controls as you can (without totally screwing up the user experience), can make it hard enough to drive attackers elsewhere for lower-hanging fruit. Regardless of how good defense in depth sounds, the reality is that with today's mobility we need to continue protecting the endpoints, as we generally can't control the location or network used by those endpoints. Obviously no one honestly believes current endpoint protection approaches work well, so it's time to evaluate how to do it better, which we will do in this paper. But let's look at our changing requirements before we dismiss existing endpoint security controls.

Control Lost

Sensitive corporate data has never been more accessible. Between PCs and smartphones and cloud services (Salesforce.com, Jive, Dropbox, etc.) designed to facilitate collaboration, you cannot assume any device – even those you own and control – isn't accessing critical information. Just think about how your personal work environment has changed over the past couple years. You store data somewhere *in the cloud*. You access corporate data on all sorts of devices. You connect through a variety of networks, some 'borrowed' from friends or local coffee shops.

We once had control of our computing environments, but that's no longer the case. You can't assume anything nowadays.

We once had control of our computing environments, but that's no longer the case. You can't assume anything nowadays. The device could be owned by the employee; and your CFO's kid could surf *anywhere* on Dad's corporate laptop. Folks connect through hotels and other public networks, and you have no idea what is lurking out there. Obviously you cannot just give up and forget about controlling your internal networks. But you know your perimeter defenses, with their fancy egress filtering and content analysis, are sometimes simply out of the picture.

An just in case the lack of control over the infrastructure isn't unsettling enough, you still need to consider the *user factor*. That unfortunate tendency of employees to click pretty much anything that looks interesting. Promiscuous clicking opens up employees to all sorts of bad stuff, bringing infections back into your corporate environment and putting your data at risk. So we need to fortify the endpoints as much as possible and assume the worst.

Advancing Adversaries

Attackers aren't making things any easier. Today's professional malware writers have gotten ahead of these trends by using advanced malware (remote access trojans [RATs] and other commercial malware techniques) to defeat traditional endpoint defenses. It is well established that traditional file-matching approaches (on both endpoints and mail & web gateways) no longer effectively detect these attacks due to techniques such as polymorphism, malware droppers, and code obfuscation.

> Vendors use the generic term "zero day" for malware you haven't seen, but the sad reality is you haven't seen *anything* important that's being launched at you. **It's all new to you.**

Even better, you cannot expect to see an attack before it hits you. Whether it's a rapidly morphing malware attack or a targeted attempt, yesterday's generic sample gathering processes (honeynets, WildList, etc.) don't help, because malware files are unique and customized to a target. Vendors use the generic term "zero day" for malware you haven't seen, but the sad reality is you haven't seen *anything* important that's being launched at you. It's all new to you.

When we said *professional* malware writers we weren't kidding. The bad guys now take an agile software approach to building their attacks. They have tools to develop and test the effectiveness of their malware and are able to determine whether existing malware protection tools will detect their attacks. Even with reputation systems and other mechanisms for detecting these advanced attacks, today's 'solutions' are just not effective enough. All this means security practitioners need new tactics for detecting and blocking malware which targets their users.

Evolving Endpoint Malware Detection

The good news is that endpoint security vendors recognized their traditional approaches were about as viable as dodo birds a few years back. They have been developing improved approaches – the resulting products have reduced footprints requiring far less computing resources on the device, and are generally decent at detecting simple attacks. But as we have described, *simple* attacks aren't the ones to worry about. So we will investigate how endpoint protection will evolve to better detect and hopefully block the current wave of attacks.

We will start by identifying behavioral indicators of a malware attack. Like a poker player, every attack includes its own 'tells' that enable you to recognize bad stuff as it happens. Then we will describe some additional data sources that can provide the context needed to determine whether something suspicious actually *is* bad. We will evaluate a number of different controls to block these attacks at different points along the attack chain. Finally we will wrap up with a candid discussion of the trade-offs and compromises involved in dealing with this advanced malware. You *can* stop these attacks, but the cure might be worse than the disease. We will offer suggestions for finding the equilibrium point between detection, response, and user impact.

Behavioral Indicators

Attackers continue to advance their tactics. Advanced attackers rarely use the same file or malware delivery vehicle twice and morph the malware files constantly, making it very hard to use the basic file-based detection which underpins traditional anti-malware tools. So efforts to detect malware can no longer focus exclusively on what malware looks like (a file hash or some other identifying factor) and must incorporate a number of new data sources for identification.

> The industry has made a tremendous research investment in profiling the kinds of behavior which indicate attacks, and in building detection tools to look for those kinds of *behavioral indicators* in real time.

These new sources include *what it does,* how it gets there, and who sent it; in combination with traditional file analysis, this broader information base enables you to improve accuracy and reduce false positives. No, we don't claim there is no place for traditional anti-malware (signature matching) any more. First of all, compliance continues to mandate AV, so unless you are one of the lucky few without regulatory oversight you don't have a choice. But more pragmatically not all attacks are 'advanced'. Attackers looking for the path of least resistance use known malware kits, leveraging known bad files. There is no reason to ever let a recognizable bad file execute on your device – certainly not just to confirm it's bad, thus AV is useful in this case. Though to be clear, existing malware engines have a blind spot if the malware kit generates polymorphic files.

But obviously the old tactics for detecting malware aren't effective at dealing with advanced and targeted attacks. These additional data sources provide additional information to help identify good and bad code more accurately, and the most promising is behavioral analysis. The good news is that the industry has made a tremendous research investment in profiling the kinds of behavior which indicate attacks, and in building detection tools to look for those kinds of *behavioral indicators* in real time as code executes on devices.

Profiling Behaviors

When we say "malware profile", what are we talking about? That depends on what you are trying to accomplish. One use for profiles is malware analysis, described in depth in Malware Analysis Quant. In this case the goal is to understand what the malware looks like and does, in detail. You can then use the profile to find other devices that have been compromised.

Another use case leverages profiles of typical malware actions to detect an attack on a device *before* infection. This is all about figuring out what the malware does and when, and then using that information to stop it *before* it does damage. Several things are useful to know for detection:

- Registry settings
- Processes/services
- Injected code
- New executables
- Domains/protocols
- Network communication targets (C&C)

Mandiant's term, Indicators of Compromise, sums it up pretty well. If the malware injects malicious code into a standard operating system file such as `winlogon.exe` or `services.exe` in Windows, adds certain registry keys to a Windows device to ensure persistence, contacts external servers known to distribute malware, or even uses an opaque encrypted protocol (presumably command and control traffic), you have useful evidence that executable is malicious and can block it.

Finite Ways to Die

Malware profiles are terrific if you can capture a sample of the malware and run it through a battery of static and dynamic analyses to really figure out what it does. But what happens if you can't get the malware? Do you just wait until devices have been owned to develop a profile? That sounds a lot like the reactive approach the industry has relied on for years – to disastrous effect.

You need a list of generic behaviors that indicate malicious activity, to use as an early warning system for possible attacks. Of course, relying purely on specific behaviors can result in false positives – because injecting code and changing registry settings *can* be legitimate actions, such as when patching. You probably learned that lesson the hard way when using host intrusion prevention technologies (HIPS) years ago. So you need to use behavioral indicators for first-level alerting, and then additional analysis to figure out whether you are *really* under attack.

But what happens if you can't get the malware? Do you just wait until devices have been owned to develop a profile? That sounds a lot like the reactive approach the industry has relied on for years – to disastrous effect.

This process is akin to receiving an alert from your SIEM. You cannot assume a SIEM alert represents an attack, but it provides a place to begin investigation. A skilled analyst examines the alert and validates or dismisses the attack, as documented in Network Security Operations Quant. How does the analyst determine whether the attack is real? By applying their experience to understand the alert's **context**.

But on a typical endpoint or server device you don't have a skilled human analyst to wade through all the potential alerts. So you need a tool which can apply sufficient context to determine what is an attack and what is not – determining what to block and what to allow.

Typical Behavioral Indicators

As we discuss the behavioral indicators typically exhibited by malware, the indicators fall into two general buckets: those which indicate compromise during an attack, and those which established malware exhibits when stealing data.

This first set of indicators is widely used by traditional endpoint protection for "behavioral heuristics" and preventing attacks.

- **Memory corruption/injection/buffer overflow:** The old standard for compromising devices is to alter the "execution flow of a program by submitting crafted input to the application." That's not our definition – it comes from Haroon Meer's 2010 paper (PDF) documenting the history of memory attacks. If you aren't familiar with this attack vector, the paper provides a great primer. Suffice it to say that memory corruption is alive and well, and any behavioral detection approach must watch for these attacks.

- **System file/configuration/registry changes:** Normal executables rarely update registry, configuration, or system file settings; so any activity of this sort warrants investigation.

- **Droppers and installing code:** Malware writers need to update their attacks faster than ever, so it's more efficient for them to plant a stub program called a dropper, which then accesses the network and downloads the latest malware files to the compromised device. So an executable that behaves like a dropper needs to be stopped. You should also be suspicious of programs that add or change executables by injecting code or making other dynamic changes as a matter of course.

- **Turning off existing protections:** A program that turns off standard security controls, such as anti-virus agents and User Account Control, is probably up to no good, so those are good malware indicators.

- **Identity and privilege manipulation:** Actions such as local account creation and privilege escalation are usually indicators of malware gaining further control of the device and/or attack other devices on the network.

This next set of indicators tends to be employed by established malware trying to steal data or further proliferate through the environment.

- **Parent/child process inconsistencies:** Some processes and executables should always be launched by specific processes and executables. Violations of these relationships might indicate malware.

- **Exploits disguised as patches:** As demonstrated so clearly by the recent Flame malware, attackers are now gaming the Windows Update process to obscure their activity. This is difficult to detect because patches are supposed to change files, inject code, and update configurations and registry settings.
- **Keyloggers:** There are few situations where a keylogger is actually legitimate application behavior, but we defer judgement for that one-in-a-million edge case, and simply point out that the presence of a keylogger or any other technique for intercepting device driver commands generally indicates bad mojo.
- **Screen grabbing:** In response to the on-screen keyboards used to defeat keyloggers, attackers also grab screens at click time to detect letters being selected. This is cumbersome but many attackers have low personnel costs (think hacker boiler rooms) so it can be economical. Screen grabbing at inappropriate times (like when logging into a banking site) is definitely something to watch for.

Of course some of these behaviors are legitimate under specific circumstances. So we reiterate the importance of context for determining whether to block or allow.

Providing Context

Detecting today's advanced malware requires more than just looking at the file like classic AV – we also need to leverage behavioral indicators. To make things more interesting, even suspicious behavior can be legitimate in certain circumstances. So for accurate and effective detection you need better context on what the code does, where it came from, and who it came from, in order to reach a reasonable verdict on whether to allow or block execution.

What happens when you don't have that context? Let's jump into the time machine and harken back to the early days of host intrusion prevention (HIPS) and HIPS-like products, which ran on devices and scanned for both attack signatures and behaviors that indicated malware. Without sufficient context, these controls blocked all sorts of things – including scads of false positives – and generally wreaked havoc on operations. That didn't work out very well for organizations which actually needed their devices up and running, even if that imposed a security cost. Go figure.

But the *concept* of watching for attacks on devices is solid. It was an implementation problem. Nowadays additional context reduces false positives, increases accuracy, and limits disruption of operations – all worthy goals for a control to manage new attack vectors. So let's dig into a few data sources beyond behavioral indicators to help identify bad stuff.

HIPS ran on devices and scanned for both attack signatures and behaviors that indicated malware. Without sufficient context, these controls blocked all sorts of things – involving scads of false positives – and generally wreaking havoc on operations.

From Where: the Dropper

We already mentioned that malware writers use *droppers* to gain a foothold on devices, and then download current and/or additional attacks, instead of attempting to get the entire malware onto the device as part of the initial compromise. Of course droppers are malware just as much as anything else, but they morph more frequently, which makes initial detection difficult. And as we described in Malware Analysis Quant, the only thing worse than being infected is getting *re-infected* by known malware.

So profiling malware droppers enables you to search for these files in your environment. By tracing the path of droppers you can identify devices that have been compromised but not yet activated. The key is analysis of data about which files are on which devices; when a file is discovered to be bad, if you have the data and analytics in place, it becomes easy to determine which devices have the bad file installed.

Of course this is still reactive. But the presence of a dropper (or similar known bad file), combined with any other bad behavior, is fairly damning evidence of a compromised device. Tracing the droppers back far enough points you to the origination point of the malware; eliminate any vestiges, and you can prevent reinfection.

What Are You Doing (and Why)?

Some obvious activities fail the "sniff test" of what is right and wrong. But malware writers have masked their malicious intentions within what appears to be acceptable functionality. For instance:

- **Browser Plug-ins:** It's pretty cool that allowed plug-ins (Skype, for example) can highlight phone numbers on standard web pages to make them easier to click and dial automatically. But what if the same technology was used to highlight interesting text in order get unsuspecting end users to click on link and get infected by a drive-by download?
- **Altering application functions:** A common technique attackers use in advanced malware is to add functionality to an application, so it's important to have a profile of the common activities of each application and then look for behavior outside of that profile. It's similar to a white listing approach, but as opposed to tracking executables, you are tracking application behavior.

Of course it's non-trivial to collect the context of profiling each application the user could use, track what they are doing in that application, and determine whether it's legitimate. But that kind of analysis is required to detect the advanced attacks we face on a daily basis.

When?: Timing the Attack

It's not enough to evaluate what the code is doing to the device, it's also important to have the context of **when**. For example, let's use screen capture as a metaphor for the importance of time to establishing context. Every operating system has the capability to capture the screen, so grabbing the screen shouldn't raise alarms. But what if the screen capture happens right as the user is clicking on a virtual keyboard logging into their bank account? Or if they are logging into their VDI (virtual desktop) interface? Right, that's probably no good. But if it happens when the user is browsing on a news site, it's probably OK. Again, you won't always be right (as the news site could be compromised), but it's about minimizing the likelihood you miss something by using additional data to provide context.

> Initially developed to improve the effectiveness of anti-spam gear, reputation has emerged as a fundamental aspect of every vendor's threat intelligence offering.

Who Dat?: Reputation

The other useful source for detecting advanced malware is the *reputation* of a file, sender, or IP address. Initially developed to improve the effectiveness of anti-spam gear, reputation has emerged as a fundamental aspect of every vendor's threat intelligence offering. The larger security vendors have access to considerable amounts of data from hundreds of millions of installed endpoints and network devices; they mine their datasets to determine which files, devices, and network addresses tend to do bad things.

This is all an inexact science – especially in light of the simplicity of morphing a file, spoofing an IP address, or fiddling with a device fingerprint. We must expect advanced adversaries to masquerade as something innocent to obfuscate their intentions. You cannot afford to rest your malware-or-clean verdict strictly on reputation – but you can use it as a supporting data source of additional context when analyzing possible attacks.

Of course malware writers don't make it *easy* to figure out what they are doing. Your best bet is to assemble as much data as you can, analyze what's going on within the device (behavioral analysis), and combine that with data from outside sources to judge the nature and intent of code running (or attempting to run) on your devices – this at least gives you a fighting chance. So far we have focused on analysis and detection, but detection doesn't help without a mechanism to actually *block* detected attacks.

Controls

Let's take our discussion of detecting advanced malware to the next level: doing something with the information we have gathered. We need to produce a *verdict* on whether something is malware or not — and if it is to block it. This is where you need to understand the trade-offs between different controls and choose the best for your environment.

The Malware Detection 'Cocktail'

Let's jump back in the time machine, to the good old days on the cutting edge of spam detection. Spammers got pretty good and evolved their techniques to evade every new defense the email security folks came up with. 3-4 years in, around 2004-2005, the vendors used 15-20 different tactics to determine whether any particular email message was unsolicited. Sound familiar? Malware detection has reached a similar point. Lots of techniques, none foolproof, and severe consequences for false positives.

What can we learn from how the anti-spam vendors evolved? Aside from the fact that the effectiveness you can achieve and maintain over time is limited? The best approach for dealing with a number of different detection techniques is to use a *cocktail* approach. This involves scoring each technique (possibly quite coarsely), feeding it into an algorithm with appropriate weighting for each technique, and determining a threshold that indicates something bad. Obviously the secret sauce is in the algorithm, which is the vendor's responsibility.

Yes, a lot of this happens (and should remain) behind the curtain and you won't have the ability to configure the algorithm or the cocktail, but we are trying to explain how the process works so you can knowledgeably evaluate new devices and products that claim to detect advanced malware.

> The best approach for dealing with a number of different detection techniques is to use a *cocktail* approach. Scoring each technique, applying an appropriate weighting for each technique, and then determining a threshold that indicates something bad.

But we know you cannot be right every time. So it's time to plug our research on incident response and forensics, including Incident Response Fundamentals, React Faster and Better, and Network Security Analysis, to ensure you are prepared for the inevitable failures of even the best malware detection.

Let's look at the components and controls you will rely on to block these detected attacks.

Traditional Endpoint Protection

Thanks to compliance mandates and check-box-centric auditors, you still need endpoint protection – often called anti-virus. But most endpoint security suites encompass much more than traditional anti-virus signatures, including some of the tactics we have discussed. Obviously with 15-20 players remaining in this market, the quality of detection is all over the map and quite dynamic. Each vendor goes through ups and downs in detection effectiveness.

So how do we recommend choosing an endpoint suite? That could be an entire series itself, but suffice it to say that the effectiveness of detection probably shouldn't be the most important selection criteria. It is too hard to verify, and they each do a decent job of finding known malware but a mediocre job of finding the advanced attacks this paper is focused on. You need endpoint protection for compliance so you should minimize price, ensure that agents can be effectively managed (especially if you have thousands of endpoints), and make sure they are as thin as possible. It's bad enough having to use a control that doesn't work as well as it needs to but crushing device performance adds insult to injury. By all means, check the latest comparative effectiveness rankings, but understand they go out of date pretty quickly.

> The earlier you can detect malware and block it, the less mess you will inevitably have to clean up.

Network-based Malware Detection

The earlier you can detect malware and block it, the less mess you will inevitably have to clean up. That means working to eliminate attacks at the perimeter or even in the cloud before an attack ever gets near your desktop. How can you do this? A new type of network security device scrutinizes ingress traffic to detect malware files before they enter your corporate network. We expect this capability to become a feature of pretty much every perimeter device over time, but for now you will need to deal with specialist companies and separate devices. We published some research on this earlier this year so check out Network-based Malware Detection for details on the approaches, limitations, and roles of these devices in your network security strategy.

Advanced Endpoint Controls

We all understand that traditional endpoint security suites leave too much attack surface exposed to advanced attackers, depending on your pain threshold (how likely you are to be targeted by an advanced attacker). An additional level of endpoint protection may be necessary. So let's discuss some of these alternatives which detect and block based on behavioral indicators, track file trajectories and proliferation, and/or allow authorized executables.

The first category of advanced endpoint control works at the application layer to provide protection for running applications at a low level within the operating system. A few new offerings have emerged leveraging the kind of malware detection cocktail discussed above. This analytical approach to what's happening on the endpoint, combined with context from application activity and behaviors (as described in Providing Context above) can reduce false positives and improve effectiveness. These tools impact user experience by blocking things (which is usually a good thing), but need to be put through proper diligence before broad deployment. But you do that with all new technologies anyway, right?

We have already talked about how malware proliferation analytics can be very useful for tracking the spread of malware within your environment, securing the origin point, and reducing the possibility of reinfection. We advocate this kind of analysis as another layer of defense. You have two main options for gathering the information for this analysis: either on the endpoint or within the network. Endpoint solutions provide a thin agent which sends information up to a cloud repository which provides analytics and visualization. leveraging outbreak data from many other organizations. Obviously this involves another agent on the desktop and another interface to manage, but community analysis can yield interesting information.

> This analytical approach to what's happening on the endpoint, combined with context from application and specific behavior can reduce false positives and improve effectiveness.

You can also look for C&C connections on your network by monitoring egress traffic. Services maintain lists of C&C networks and patterns of communication used by botnets to identify compromised devices participants. Keep in mind that this is a step late – the device is already compromised by this point – but can produce an accurate assessment of which devices need to be cleaned up *immediately,* as they are known to be behaving badly. Unfortunately looking at the network does not provide definitive identification of the malware origin point, which limits its utility for reducing reinfection.

Finally you have a draconian option: application whitelisting. This involves a "default deny" approach on endpoints, which only allows a set of authorized executables to run on the protected endpoints, blocking everything else. This is draconian because it dramatically impacts the user experience – generally not in a good way. Most whitelisting products offer grace periods to allow execution of programs until an administrator approves or rejects the request, but this compromise violates the security model. Some vendors perform memory analysis and are introducing other behavioral approaches to make the grace period less risky, but a grace period inherently introduces significant risk. We see AWL as more appropriate to fixed function devices such as kiosks, call centers, control systems, etc., where general purpose software shouldn't be running.

> Most of these *advanced* tactics will eventually be subsumed into existing controls, either via acquisition or internal development. That's just how security markets (and most other technology markets) work.

Of course most of these *advanced* tactics will eventually be subsumed into existing controls, either via acquisition or internal development. That's just how security markets (and most other technology markets) work. What's advanced today will be standard tomorrow. But the process can take 2-3 years, and most organizations cannot afford to wait, so you can evaluate many of these technologies to fill the gap.

Of course not all these controls run exclusively on endpoints. Despite the title of this series, you need to use a variety of the controls at your disposal, and some work better at other places within your IT infrastructure. Weighing these trade-offs and designing an effective, layered control set is the art, as opposed to the science of information security.

Trade-offs and Compromises

Time

To detect advanced malware you need to include *time* in your planning. Different approaches are more or less effective depending on when you use them. For instance, the reputation of a sender or file is most valuable early. If you get a hit with reputation-based intelligence, you can skip other more demanding analyses. Likewise, malware file signature checks are quick and should be performed early.

But it is getting much harder to detect attacks *before* malware executes. Many of the behavioral indicators we have mentioned are only available when the malware is running, and others appear once it has activated. This chart shows what we mean.

Detection Before/During/After Attacks

BEFORE THE ATTACK	DURING THE ATTACK	AFTER THE ATTACK
Reputation (Device, Sender)	Memory corruption/buffer overflow	Egress network traffic
File signatures	System files/configuration/registry changes	Command and Control network traffic
Network-based malware detection	Parent/child process inconsistencies	Malware file proliferation analysis
Email/Phishing defense	Droppers and installing code	Keylogger activity
Web filtering	Turning off security controls (like AV and/or HIPS)	Screen grabbing
		Privilege escalation (and other fun identity attacks)
		DLP (and other content filtering techniques)

Obviously the earlier you can detect the attack, the better. So controls that detect malware *before* any potential infections are preferable. But as we have mentioned repeatedly, it is increasingly difficult to detect

advanced attacks until before malware runs, which brings both the during and after periods into play for detection. As with most security, the right answer is *all of the above,* mixing and matching controls at all steps of the attack chain to maximize your chance of detecting the attack and subsequent compromise.

Device/Location Variance

Another aspect to consider when designing control sets is the amount of control you have over the device, and what kind of device it is. There is substantial variation in what you can do to protect mobile devices and PCs, especially if your corporation doesn't own the device (think BYOD). Obviously you have the most control over corporate PCs where you can perform a credentialed scan, install a device agent to check file signatures and reputation, and check for behavioral indicators.

There is substantial variation in what you can do to protect mobile devices and PCs, especially if your corporation doesn't own the device (think BYOD).

On devices you don't control, such as those belonging to contractors and customers and perhaps employees, you might be able to scan at connection to the network or install a browser plug-in to protect a specific web application or set of domains. But you need to tread carefully – privacy is often a major concern on devices you don't control. If installing an agent or plug-in is a non-starter you need to rely on the before-attack controls described above to (hopefully) prevent the attack when the device is connected to your network. Though after-attack network monitoring should be used as a fallback to catch attackers misbehaving *before* they exfiltrate your data.

When the device is connected to your network it gains ingress and egress protection from your perimeter security controls. When it's not connected, these network-based controls are unavailable.

Smartphones are a bit different. You may be able to install an agent but functionality varies widely between the various mobile operating systems. Don't expect much ability to check behavioral indicators on a smartphone, as agents rarely have real-time access to mobile kernels. Android provides more access than iOS, so anti-malware agents are available on Android. But any mobile agent is quite limited compared to a PC-based agent. Unless the mobile device is jailbroken or rooted, which is a whole different discussion.

Remember that the current location of a device affects your ability to protect it. When the device is connected to your network (whether physically or via VPN) it gains ingress and egress protection from your perimeter security controls. You can block attacks at the perimeter via network-based malware detection or email security devices. You can also perform egress filtering to check for C&C traffic or data exfiltration, which indicate an attack.

When the device is not connected to your network those network-based controls are unavailable. So you need some type of agentry for a fighting chance – and to scrutinize the device when it connects to your network, just to make sure nothing bad happened to the device while it was *out in the wild*.

Compromises

The controls at your disposal range from monitoring to locking down devices. Detecting advanced malware requires all of them, but you need to be conscious of disruptive impact on end users. Find a balance that is sufficiently secure but not too disruptive, navigating the constraints of device ownership and control, and workable across device locations and network connectivity scenarios. There is no simple right answer – just an opportunity to manage expectations and ensure that decision makers understand the compromises they choose.

Summary

Today's adversaries have significant funding, expertise, and the patience to continue poking and prodding their targets until they gain a foothold. That first foothold is then used to establish a "base camp" of sorts within your environment to further attack devices, steal data, and maintain presence in your operations.

But one of the misnomers of this class of attackers is that they always use advanced malware and targeted attacks. *The attackers only use as advanced an attack as they need to.* So if an organization has limited defenses, attackers use simplistic approaches. Against organizations that invest in strong network and application controls, implement a precise security program, and monitor much of the activity within their environments, attackers need to use very sophisticated attacks to achieve their objectives.

The detection techniques employed by most endpoint protection offerings are insufficient to catch or block these advanced attacks so the industry needs to move beyond traditional detection and increase the sophistication of the approaches used to combat these attackers. Tactics such as behavioral analysis, reputation, and malware proliferation analytics provide a more robust foundation to find attacks before data is lost.

Of course without context to factor in what the user is doing you are destined to repeat the failures of technologies like host intrusion prevention (HIPS), which threw off many false positives and adversely disrupted technology operations. Other controls, including application whitelisting, provided protection from these advanced attacks but at the cost of adversely disrupting the user experience. Ultimately organizations have decided that it is more important not to impact users adversely than to protect them from malware, so security professionals need to go back to the drawing board to improve detection without disrupting the user experience.

Traditional detection techniques still have a place finding attacks you have already seen. But to detect advanced attacks and protect against determined attackers you need a combination of advanced endpoint analysis techniques to provide the sufficient context to determine whether a particular behavior is good or bad *at that moment*.

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

About the Analyst

Mike Rothman, Analyst/President

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

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

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

About Securosis

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

Our services include:

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

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

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

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

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

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

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

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