



Endpoint Security Fundamentals

Version 2.1

Released: June 2, 2010

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 Lumension Security



Lumension Security, Inc., a global leader in endpoint management and security, develops, integrates and markets security software solutions that help businesses protect their vital information and manage critical risk across network and endpoint assets. Lumension enables more than 5,100 customers worldwide to achieve optimal security and IT success by

delivering a proven and award-winning solution portfolio that includes vulnerability management, endpoint protection, data protection, antivirus and reporting and compliance offerings. Lumension is known for providing world-class customer support and services 24x7, 365 days a year. Headquartered in Scottsdale, Arizona, Lumension has operations worldwide, including Virginia, Texas, Utah, Florida, Ireland, Luxembourg, the United Kingdom, Australia, and Singapore. Lumension: IT Secured. Success Optimized.™ More information can be found at www.lumension.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

Introduction	3
Prioritize: Finding the Leaky Buckets	5
Triage: Fixing the Leaky Buckets	7
Controls: Update and Patch	9
Controls: Secure Configurations	11
Controls: Anti-Malware	13
Controls: Firewalls, HIPS, and Device Control	16
Controls: Full Disk Encryption	18
Building the Endpoint Security Program	20
Endpoint Incident Response	23
Endpoint Compliance Reporting	25
Who We Are	27
About Securosis	28

Introduction

Coming from the “sad but true” files is the reality that many folks have just given up on protecting the endpoint. Yes, we all go through the motions of having endpoint agents installed (on Windows anyway), but most of us have pretty low expectations for anti-malware solutions. Justifiably so, but that doesn’t mean it’s game over. There are plenty of things we can do to better protect the endpoint, some of which were discussed in the Securosis post [Low Hanging Fruit: Endpoint Security](#).

But let’s not get the cart ahead of the horse. Nowadays there are lots of incentives for the bad guys to control endpoint devices. Private data typically resides on the endpoint, including nice things like customer databases — and with the strategic use of a keylogger, it’s just a matter of time before bank passwords are discovered. And let’s not forget about intellectual property on the devices, since far too many employees have their deepest darkest (and most valuable) secrets on their laptops, within easy reach. Best of all, compromising an endpoint device gives the bad guys a foothold in an organization, and enables them to compromise other systems and spread the love.

The endpoint has become the path of least resistance, mostly because of the unsophistication of the folks using said devices doing crazy Web 2.0 stuff. All that information sharing certainly seemed like a good idea at the time, right? Regardless of how wacky the attack, it seems at least one clueless user will fall for it. Between web application attacks like XSS (cross-site scripting), CSRF (cross-site request forgery), social engineering, and all sorts of drive-by attacks, compromising devices is like taking candy from a baby. But not all the blame can be laid at the feet of users, because many attacks are pretty sophisticated, and even hardened security professionals periodically fall for these attacks.

Combine that with the explosion of mobile devices, whose owners tend to either lose them or bring back bad stuff from coffee shops and hotels, and you’ve got a wealth of soft targets. As the folks tasked with protecting corporate data and ensuring compliance, we’ve got to pay more attention to locking down the endpoints — to the degree we can. That’s what the Endpoint Security Fundamentals series is all about.

Philosophy: Real-world Defense in Depth

As with all of Securosis’ research, we focus on tactics to maximize impact for minimal effort. In the real world, we may not have the ability to truly lock down the devices since those damn users want to do their jobs. The nerve of them! So we’ve focused on layers of defense, not just from the standpoint of technology, but also looking at what we need to do before, during, and after an incident.

- **Prioritize:** This will warm the hearts of all the risk management academics out there, but we do need to start the process by understanding which endpoint devices present the most risk because they hold valuable data — for a legitimate business reason, right?
- **Triage:** Once we know what’s important, we need to figure out how porous our defenses are, so we’ll be assessing the endpoints. Then we have to put a plan in place to address the issues that represent clear and present dangers.

- **Focus on the fundamentals:** Next up, we actually pick that low hanging fruit, and do the things that we should be doing anyway. That means things like keeping software up to date, leveraging what we can from malware defense, using new technologies like personal firewalls and HIPS (yes, that was sarcasm) and protecting the data with full disk encryption. Right, none of this stuff is new, but not enough of us do it.
- **Building a sustainable program:** It's not enough to just implement some technology. We also need to do some of those softer management things which we don't like very much — like managing expectations and defining success. Ultimately we need to make sure the endpoint defenses can (and will) adapt to the changing attack vectors we see.
- **Respond to incidents:** Yes, it will happen to you, so it's important to make sure your incident response plan accounts for the reality that an endpoint device may be the primary attack vector. So make sure you've got your data gathering and forensics kits at the ready, and also have an established process for when a remote or traveling person is compromised.
- **Document controls:** Sorry, but we have to mention the word 'compliance' in any kind of security research. The auditor will show up and want to know what controls you have in place to protect those endpoints. So for all those fancy (and effective) controls in place, you also need to focus on documentation, ensuring you can substantiate all the tactics implemented.

Prioritize: Finding the Leaky Buckets

Since hope is not a strategy (not for security, anyway), you can't just make assumptions about what's installed, what's configured correctly, and what the end users actually know. You need to take a structured approach to figuring that out, which involves using some of the same tactics our adversaries use against us.

The goal here is twofold: first figure out what presents a clear and present danger to your organization, and put a triage plan in place to remediate those issues. Second, manage expectations at all points in this process. That means documenting what you find (no matter how ugly the results) and communicating that to management, so they understand the situation.

To be clear, although we focus on endpoint security here, this prioritization (and triage) process makes up the first steps in any security program.

Assessing the Endpoints

In terms of figuring out your current state, you need to pay attention to a number of different data sources — all of which yield information to help you understand the current state.

- **Endpoints:** Yes, the devices themselves need to be assessed for updated software, current patch levels, unauthorized software, etc. You may have a bunch of this information in a patch/configuration management product or as part of your asset management system. To confirm that data, we'd also recommend you let a vulnerability scanner loose on at least some of the endpoints, and play around with automated penetration testing software to check for exploitability of the devices.
- **Users:** If we didn't have to deal with those pesky users, life would be much easier, eh? Well, regardless of the defenses you have in place, an ill-timed click by a gullible user and you are pwned. You can test users by sending around fake phishing emails and other messages with fake bad links. You can also distribute some USB keys and see how many people actually plug them into machines. These "attacks" will determine pretty quickly whether you have an education problem and what other defenses you may need to overcome those issues.
- **Data:** Despite our endpoint focus, we still need to figure out where the sensitive data resides and that means a discovery process. You need to identify devices with sensitive information since those warrant a higher level of protection. Perhaps you can leverage other internal efforts to do data discovery, but regardless, you need to know which devices would trigger a customer disclosure if lost/compromised.

- **Network:** Clearly compromised devices need to be identified and remediated *quickly*. The network provides plenty of information to identify compromised devices. Whether it's looking at network flow data, anomalous destinations, or alerts on egress filtering rules — the network is a pretty reliable indicator of what's already happened, and where your triage efforts need to start.
- **Policy:** This is not really data centric, but many security issues result from the lack of a policy, or from inadequate communication of policy. So now is a good time to revisit your email and web usage policies and make sure everyone is on the same page regarding consequences for violating those policies.

Keep in mind that it is what it is. You'll likely find some pretty idiotic things happening, or confirm idiotic things you already knew about. The point isn't to get overwhelmed, it's to figure out how much is broken so you can start putting in place a plan to fix it, and then a process to make sure it doesn't happen so often.

Prioritizing the Risks

Prioritization is more art than science. After spending some time gathering data from the endpoints, users, data, and network, how do you know what to focus on first? Not to be trite, but it's basically a common sense thing.

For example, if your network analysis showed a number of endpoints already compromised, start by fixing those. Likewise, if your automated pen test showed you could get to a back-end datastore of private information via a bad link in an email (clicked on by an unsuspecting user), then you have a clear and present danger to deal with, no?

After you douse the hottest fires, prioritization gets down to who has access to sensitive data and making sure those devices are protected. This sensitive data could be private data, intellectual property, or anything else you don't want to see on the full-disclosure mailing list. Hopefully your organization knows what data is sensitive, so you can figure out who has access to that data and build the security program around protecting that access.

In the event there is no internal consensus about what data is important, you can't be bashful about asking questions like, "Why does that sales person need the entire customer database?" and "Although it's nice that the assistant to the assistant controller's assistant wants to work from home, should he have access to the unaudited financials?" Part of prioritizing risk involves identifying idiotic access to sensitive data.

Many security folks believe they know what is important and don't bother asking business users, but don't fall into that trap. The opinion of the security team is interesting, but ultimately irrelevant to the prioritization function. The business leaders need to make the call on priorities. It's their money and probably their jobs at risk if data is compromised.

Finally, every business user believes their issues are most important. Keep in mind that not everything can be a Priority 1. So there will likely be a contentious debate where all the senior leaders need to get in a room and reach consensus about priorities across the organization.

Jumping on the Moving Train

In the real world, you don't get to stop everything and start your security program from scratch. You've already got all sorts of assessment and protection activities going on — at least we hope you do. That said, we do recommend you take a step back and not be constrained to existing activities. Existing controls provide input to your data gathering process, but you need to think bigger about the risks to your endpoints and design a program to handle them.

Triage: Fixing the Leaky Buckets

The next step in our endpoint security journey involves building, communicating, and executing on a triage plan to fix those *leaky buckets*. The plan consists of the following sections: Risk Confirmation, Remediation Plan, Quick Wins, and Communication.

Risk Confirmation

Coming out of the prioritize step, before we start committing resources and/or yanking at the fire alarm, let's take a deep breath and make sure our ranked list really represents the biggest risks. How do we do that? Basically by using the same process we used to come up with the list. Start with the most important data, and work down the issues we've already found.

The best way to get everyone on the same page is to have a streamlined meeting between the key influencers of security priorities. That involves folks not just within the IT team, but also probably some tech-savvy business users — since it's their data at risk. Yes, we are going to go back to them once we have the plan. But it doesn't hurt to give them a heads up early in the process about what the highest priority risks are, and get their buy-in early and often throughout the process.

Remediation Plan

Now comes the fun part: we have to figure out what's involved in addressing each of the leaky buckets. That means figuring out whether you need to deploy a new product, optimize a process, or both. Keep in mind that for each of the discrete issues, you want to define the fix, the cost, the effort (in hours), and the timeframe commitment to get it done. No, none of this is brain surgery, and you probably have a number of fixes on your project plan already. But hopefully this process provides the needed incentive to get some of those projects moving.

Once the first draft of the plan is completed, start lining up the project requirements within the realities of budget and availability of resources. When it comes time to present the plan to management (including milestones and commitments), you have already had the visit with Mr. Reality so you understand what is feasible and can manage expectations accordingly.

Quick Wins

As you are doing the analysis to build the remediation plan, it'll be obvious that some fixes are cheap and easy. We recommend you take the risk (no pun intended) and take care of those issues first, regardless of where they end up on the priority list. Why? We want to build momentum behind the endpoint security program (or any program, for that matter) and that involves showing progress as quickly as possible. You don't need to ask permission for everything.

Communications

The hallmark of any pragmatic security program (read more about the [Pragmatic philosophy here](#)) is frequent communications and senior level buy-in. So once we have the plan in place, and an idea of resources and timeframes, it's time to get everyone back in the room to get thumbs up for the triage plan.

You need to package up the triage plan in a way that makes sense to the business folks. That means thinking about business impact first, reality second, and technology probably not at all. These folks want to know what needs to be done, when it can get done, and what it will cost.

We recommend you structure the triage pitch roughly like this:

- **Risk Priorities:** Revisit the priorities everyone has hopefully already agreed to.
- **Quick Wins:** Go through the stuff that's already done. Seeing progress is already being made will usually put the bigwigs in a good mood.
- **Milestones:** These folks don't want to hear the specifics of each project. They want the bottom line. When will each of the risk priorities be remediated?
- **Dependencies:** Now that you've told them what you need to do, next tell them what constraints you are operating under. Are there budget issues? Are there resource issues? Whatever it is, make sure you are very candid about what can derail efforts and impact milestones.
- **Sign-off:** Then get them to sign in blood on what will get done and when.

Dealing with Shiny Objects

To be clear, getting to this point in the process tends to be a straightforward process. Senior management knows stuff needs to get done and your initial plans should present a good way to get them done. But the challenge is only beginning, because as you start executing on your triage plan, any number of other *priorities* will present that absolutely, positively, need to be dealt with.

In order to have any chance to get through the triage list, you'll need to be disciplined about managing expectations regarding the impact of each new shiny object on your committed milestones. We also recommend a monthly meeting with the influencers to revisit the timeline and recast the milestones — given the inevitable slippages due to other priorities.

Controls: Update and Patch

Running old software is bad. Bad like putting a new iPad in a blender. Bad because all software is vulnerable software, and with old software even unsophisticated bad guys have weaponized exploits to compromise the device. So the first of the Endpoint Security Fundamentals technical controls ensures you run updated software.

Does that mean you need to run the latest version of all your software packages? Can you hear the rejoicing across the four corners of the software ecosystem? Actually, it depends. What you do need to do is make sure your *endpoint devices are patched within a reasonable timeframe*. At least one minute before the weaponized exploit hits the grey market or shows up in Metasploit.

Assess Your (Software) Assets

Hopefully you have some kind of asset management in place, which can tell you what applications run in your environment. If not, your work gets a bit harder because the first step requires you to inventory software. It's not about license enforcement — it's about risk assessment. You need to figure out your software vendors' track records on producing secure code, and then on patching exploits as they are discovered. You can use sites like [US-CERT](#) and [Secunia](#), among others, to figure this out. Your anti-malware vendor also has a research site where you can look at recent attacks by application.

You probably hate the word *prioritize* by now, but that's what we need to do (again). Based on the initial analysis, stack rank all your applications and categorize into a few buckets.

- **High Risk:** These applications are in use by 50M+ users, thus making them high-value targets for the bad guys. Frequent patches are issued. Think Microsoft stuff (all of it), Adobe Acrobat, Firefox, etc.
- **Medium Risk:** Anything else that has a periodic patch cycle and is not high-risk. This should be a big bucket.
- **Low Risk:** Apps which aren't widely used (security by obscurity) and tend to be pretty mature, meaning they aren't updated frequently.

Before we move on to the updating/patching process, while you assess the software running in your environment, it makes sense to ask whether you really need all that stuff. Even low-risk applications provide attack surface for the bad guys, so eliminating software you don't need (or which is no longer in use) is a good thing for everyone. Yes, it's hard to do, but that doesn't mean we shouldn't try.

Defining the Update/Patch Process

Next you need to define your update and patching process — you'll have three different policies for high, medium and low risk applications. The good news is your friends at Securosis have already documented every step of this process, in gory detail, through our [Patch Management Quant](#) research.

At a very high level, the cycle is: Monitor for Release/Advisory, Evaluate, Acquire, Prioritize and Schedule, Test and Approve, Create and Test Deployment Package, Deploy, Confirm Deployment, Clean up, and Document/Update Configuration Standards. Within each phase of the cycle, there are multiple steps.

Not every step defined in PM Quant will make sense for your organization, so pick and choose what's required. The requirements are to have a defined, documented, and operational process; and to answer the following questions for each of your categories:

- Do you update to the latest version of the application? Within what timeframe?
- When a patch is released, how soon should it be applied? What level of testing is required before deployment?

In a perfect world, everything should be patched immediately and all software should be kept at the latest version. Unless you are talking about Microsoft Vista <grin>. But we all know the world isn't perfect and there are real economic and resource dependencies to tightening the patch window and buying software updates — and discovering bugs in the patches themselves, the hard way. So all these factors need to be weighed when defining the process and policies. There is no right or wrong answer — *it's a matter of balancing economic reality against risk tolerance.*

Keep in mind that patching remote and mobile users are quite different, and you have to factor that into the process. Many of these folks connect infrequently and may not have access to high-bandwidth connections. Specifying a one-day patch window for installing a 400MB patch at a mobile office in the jungle may not be realistic.

Tools and Automation

Many tools can help you automate your software updating and patching process. They range from full-fledged asset and configuration management offerings to fairly simple patching products. The nuances of configuration/patch management are beyond the scope of this series, but any organization with more than a couple hundred users needs a tool.

Controls: Secure Configurations

Next let's focus on the configurations of the endpoint devices that connect to our networks. Silly configurations present another path of least resistance for the hackers to compromise them. For instance, there is no reason to run FTP on an endpoint device, and your standard configuration should factor that in.

Define Standard Builds

Initially you need to define a *standard* build, or more likely a few. Typically for desktops (no sensitive data, and sensitive data), mobile employees, and maybe kiosks. There probably isn't a lot of value to going broader than those 4 profiles, but that will depend on your environment.

A good place to start is one of the accepted benchmarks of configurations available in the public domain. Check out the [Center for Internet Security](#), which produces configuration benchmarks for pretty much every operating system and many major applications. In order to see your tax dollars at work (if you live in the US) also consult NIST, especially if you are in the government. Its [SCAP configuration guides](#) provide similar enumeration of specific settings to lock down your machines.

To be clear, we need to balance security with usability and some of the configurations suggested in the benchmarks clearly impact usability. So it's about figuring out what will work in your environment, documenting those configurations, getting organizational buy-in, and then implementing.

It also makes sense to put together a list of *authorized software* as part of the standard builds. You can have this authorized software installed as part of the endpoint build process, but it also provides an opportunity to revisit policies on applications like iTunes, QuickTime, Skype, and others which may not yield a lot of business value and have histories of vulnerability. We're not saying these applications should not be allowed — you have to figure that out in the context of your organization — but you should take the opportunity to ask the questions.

Anti-Exploitation

As you define your standard builds, at least on Windows, you should turn on anti-exploitation technologies. These technologies make it much harder to gain control of an endpoint through a known vulnerability. I'm referring to DEP (data execution prevention) and ASLR (address space layout randomization), though Apple is implementing similar capabilities in their software.

To be clear, anti-exploitation technology is not a panacea — as the winners of Pwn2Own at CanSecWest show us every year. Especially for those applications that don't support anti-exploitation (d'oh!), but these technologies do help make it harder to exploit the vulnerabilities in compatible software.

Other Considerations

- **Running as a standard user:** We've written a bit on the possibilities running in [standard user mode \(as opposed to administrator mode\)](#), and you should consider this when designing secure configurations, especially to help enforce authorized software policies.
- **VPN to Corporate:** Given the reality that mobile users *will* do something silly and put your corporate data at risk, one technique to protect them is to run all their Internet traffic through the VPN to your site. Yes, it adds a bit of latency, but at least the traffic will be running through the web gateway and you can both enforce policy and audit what the user is doing. As part of your standard build, you can enforce this network setting.

Implementing Secure Configurations

Once you have the set of secure configurations for your device profiles, how do you start implementing them? First make sure everyone buys into their decisions and understands their ramifications. Especially if you plan to stop users from installing certain software or blocking other device usage patterns. Constantly asking for permission for things users can and can't do is a dangerous precedent, but without buy-in a device usage policy is doomed to fail. If the end users feel they need to go around the security team and its policies to get the jobs done everyone loses.

Once the configurations are locked and loaded, you need to figure out how much work is required for implementation. Assess the existing endpoints against the configurations. Lots of technologies can do this, ranging from Windows management tools, to vulnerability scanners, to third party configuration management offerings. The scale and complexity of your environment should drive the selection of the tool.

Then plan to bring those non-compliant devices into the fold. Yes, you could just flip the switch and make the changes, but since many of the configuration settings will impact user experience, it makes sense to do a bit of proactive communication to the user community. Of course some folks will be unhappy, but that's life. More importantly, this should help cut down the help desk mayhem when some things (like running that web business on corporate equipment) stop working.

Which brings us to automation. For organizations with more than a couple dozen machines, a pretty significant ROI is available from investing in some type of configuration management tool set. Again, it doesn't have to be the Escalade of products, and you can even look at things like Group Policy Objects in Windows. The point is that making manual changes to a fleet of devices is idiotic, so apply the level of automation that makes sense in your environment.

Finally, we also want to institutionalize the endpoint configurations, and that means we need to get devices built using the secure configuration. Since you probably have an operations team that builds the machines, they need to get the image and actually use it. But since you've gotten buy-in at all steps of this process, that shouldn't be a big deal, right?

Controls: Anti-Malware

At the risk of being Master of the Obvious, hopefully by this point it's clear that adequately protecting endpoint devices entails more than just an endpoint security suite. Yet, we still have to defend against malware, which means we need to figure out what is important in an endpoint suite and how to get the most value from the investment.

The Rise of Socially-Engineered Malware

Malware has dramatically changed over the past few years. Not just the techniques used, but also the volume. It's typical for an anti-virus company to identify **1-2 million** new malware samples per month. Yes, that's a huge amount. But it gets worse: a large portion of malware today is hidden within legitimate looking software.

A good example of this is fake anti-virus software. If one of your users happens to click on a link and end up on a compromised site (by any means), a nice little window pops up telling them they are infected and need to download an anti-virus program to clean up the attack. *Part* of that is true — upon visiting the site a drive-by attack did compromise the machine. But in this case the antidote is a lot worse, because this new 'anti-virus' package leaves behind a nasty trojan (typically ZeuS or Conficker).

The folks at [NSS Labs](#) have dubbed this attack "socially-engineered malware," because it hides the malware and preys on the user's good intentions to install the compromised payload, with disastrous results.

Cloud and Reputation

The good news is that the anti-malware companies are not sitting still. They continue to invest in new detection techniques to keep pace. Some do better than others (check out [NSS Labs' comparative tests](#) for the most objective and relevant testing — in our opinion, anyway), but what is clear is how broken the old blacklist, signature-based model has gotten. With 2 million malware samples per month, there is no way keeping a list of bad stuff on each device remains feasible.

The three main techniques added over the past few years to general anti-malware defense are:

- **Cloud-based Signatures:** Since it's not possible to keep a billion signatures in an endpoint agent, the vendors try to divide and conquer the problem. They split the signature database between the agent and an online (cloud) repository. If an endpoint encounters a file not in its local store, it sends a signature to the cloud for checking against the full list. This has given the blacklist model some temporary legs, but it's not a silver bullet, and the AV vendors know it.
- **Reputation:** A technique pioneered by the anti-spam companies a few years ago involves inferring the *intent* of a site by tracking what that site does and assigning it a reputation score. If the site has a bad reputation, the endpoint agent doesn't let the site's files or executables run. Some services assume that unless proven good, an IP has a bad reputation. This "guilty until proven innocent" approach can be useful, given that most bots and new sites don't have any reputation (as opposed to a bad one) and therefore would not trigger a bad reputation filter. Obviously this is highly

dependent on the scale and accuracy of the reputation database. But keep your eye on reputation technology, as it's integral to most security offerings — including perimeter and web filtering, in addition to anti-spam and endpoint security.

- **Integrated HIPS:** Another technique in use today is host intrusion prevention. But not necessarily signature-based HIPS, which was the first generation. Today most HIPS looks more like file integrity monitoring, so the agent has a list of sensitive system files which should not be changed. When a malware agent runs and tries to change one of these files, the agent blocks the request and detect the attack.

Today's anti-malware agents attempt to detect malware both before execution (via reputation) and during execution (via signatures and HIPS), so they can block attacks. But to be clear, this industry is *always* trying to catch up with the malware authors.

Making things even more difficult, users have an unfortunate tendency to disregard security warnings, resulting in devices getting compromised. As usual, our own users are often the weakest link in the chain. This can be alleviated slightly by eliminating user intervention where practical. For instance, if we *know* it's a virus, we don't have to offer the user a chance to run it or an option to clean it — it just happens. Other tactics include stronger administrative policies, which do not let users override the anti-malware software, and running as standard users (instead of admin). But it's still a fundamentally intractable problem.

Management Is Key

Selecting an anti-malware agent typically comes down to two factors: price and management. Price is obvious — plenty of upstarts want to take market share from Symantec and McAfee. They use price and an aggressive distribution channel to displace the incumbents. All the vendors also have effective migration tools, which dramatically lower switching costs.

In terms of management, it usually comes down to personal preference, because all the tools have reasonably mature consoles. Some use open data stores so customers can build their own reporting and visualization tools. Beware that time spent customizing a visualization and/or building a reporting infrastructure (that the vendor should bundle anyway) creates barriers to switching over time. But in reality the built-in visualizations, dashboards, and reports are good enough. Architecturally, some consoles are more distributed than others, and so scale better to large enterprise operations. But anti-malware remains a commodity market.

Also consider the size and frequency of signature and agent updates, especially for larger environments. If the anti-malware vendor sends 30MB updates 5 times a day, that will create problems in low-bandwidth environments such as South America and Africa.

Free AV: You Get What You Pay for...

Another aspect of anti-malware to consider is free AV, pioneered by folks like AVG and Avast, who claim up to 100 million users of their free products. To be clear, in a consumer context free AV can work fine. But it's not a suite, so you won't get a personal firewall or HIPS, though there are other free offerings for firewall and HIPS. Additionally, there won't be a cloud-based offering behind the tool, and it won't use new techniques like reputation to defend against malware. Finally, there are no management tools, so you'll have to manage every device individually for AV, which becomes infeasible past a handful.

For a number of use cases (such as your Mom's machine), free AV should be fine. And to be clear, the entire intent of these vendors in giving away the anti-malware engine is to entice you to upgrade to their paid products. That said, we use free AV on our PCs, and also in the virtual Windows images running on our Macs, and it works fine. But free AV is generally a poor fit for organizations.

White Listing: Disruptive or Niche?

You can't really talk about anti-malware without mentioning Application White Listing (AWL). This approach basically allows only authorized executables to run on endpoint devices, thus blocking unauthorized applications — which includes malware. AWL has a reputation of very disruptive to the end-user experience by breaking lots of authorized applications. Part of that is deserved, but over time we believe the technology has the potential to fundamentally change how we fight malware.

To be clear, AWL is not there yet. For some use cases such as embedded devices, kiosks, and control systems; the technology is a no-brainer. For general purpose PCs, it comes back to how much political capital the security team has for dictating what can run and what can't. Though management capabilities such as trusting certain application patches/updates, digitally signed code, and trusted locations are emerging to address the user disruption issue, we believe AWL will remain a niche technology for the next 2-3 years for general purpose malware defense.

Yet focusing on just the technology constraints doesn't tell the full story potentially hindering AWL adoption. Ultimately AWL as a technology needs to overcome the perception that it disrupts user experience and is hard to manage. As they say, perception is reality and end-user mindset needs to change for AWL to earn its place in endpoint defense. But we still believe in the potential of AWL, which will continue to mature; as the traditional methods of detecting and blocking malware increasingly fail, we expect AWL to become a key technique and appear in more and more anti-malware suites.

Layers of Defense

With all that said, we still default to the tried and true layering of security defenses. Anti-malware agents cannot be the only defense against the bad stuff out there — not if you actually want to protect your devices, anyway. We've harped on this throughout the series, regarding the importance of using other tactics on the endpoints (including [running updated software](#) and [secure configurations](#)) and within the network to compensate for the fact that anti-malware is an inexact science. And don't forget the importance of [monitoring everything](#) on your network, because as much as we try to *prevent* trouble, [reacting faster](#) is often our only option.

Controls: Firewalls, HIPS, and Device Control

Popular awareness of endpoint security revolves around [anti-malware](#). But they are called *suites* for a reason — other security components ship in these packages, which provide additional layers of protection for the endpoint. Here we'll discuss firewalls, host intrusion prevention, and USB device control.

Firewalls

We know what firewalls do on the perimeter of the network: selectively block traffic that goes through gateways by port and protocol. The functionality of a *host firewall* on an endpoint is similar. They allow organizations to enforce policy governing what traffic the device can accept (ingress filtering) and transmit (egress filtering).

Managing the traffic to and from each endpoint serves a number of purposes, including hiding the device from reconnaissance, notifying users or administrators when applications attempt to access the Internet, and monitoring exactly what the endpoints are doing. Many of these capabilities are available separately on the corporate network, but when traveling or at home and not behind the corporate perimeter, the host firewall is the first defense against attacks.

Of course a host firewall (like everything else that runs on an endpoint) takes up resources, which can be a problem on older or undersized machines. It is also important to remember that alerts multiply, especially when you have a couple thousand endpoints forwarding them to a central console, so some kind of automated alert monitoring becomes critical.

Although pretty much every vendor bundles a host firewall with their endpoint suite nowadays, the major operating systems also provide firewall options. Windows has included a firewall since XP, but the XP firewall does not provide egress (outbound) filtering — an issue remedied with Windows Vista. Mac OS X 10.5 Leopard added a 'socket' firewall to manage application listeners (ingress), and deprecated the classic [ipfw](#) network firewall, which is still included.

As with all endpoint capabilities, just *having* the feature isn't enough, since the number of endpoints to be managed puts a focus on managing the policies. This makes *policy management* more important than firewall engine details.

Host Intrusion Prevention Systems (HIPS)

We know what network intrusion detection/prevention products do, in terms of inspecting network traffic and looking for attacks. Similarly, host intrusion detection/prevention capabilities look for attacks by monitoring what's happening on the endpoint. This can include application behavior, activity logs, endpoint network traffic, system file changes, Windows registry changes, processes and/or threads, memory allocation, and pretty much anything else.

The art of making host intrusion prevention work is in setting up the policies to prevent malware infection, without badly impacting the user experience or destroying the signal-to-noise ratio of alerts coming into the management console. Yes,

this involves tuning, so start with the product's default settings (hopefully on a test group) and see what works and what doesn't. You should be able to quickly optimize the policy.

Given the number of applications and activities at each endpoint, you can go nuts trying to manage these policies, which highlights the importance of [standard builds](#). Start with 3-4 different policies, and then you can manage others by exception. Keep in mind that tuning the product for servers is totally different, as the policies will need to be tailored for each applications/server configuration, rather than a few standard images.

Currently, all the major endpoint suites include simple HIPS capabilities. Some vendors also offer more capable HIPS products — typically targeting server devices, which are higher profile targets and subject to different attacks.

USB Device Control

Another key attack vector for both data compromise and malware proliferation is the USB ports on endpoint devices. In the old days, you'd typically know when someone brought in a huge external drive to pilfer data. Nowadays many of us carry a 16GB+ drive at all times (yes, your smartphone is effectively a fancy thumb drive), so we must control USB ports to address this exposure.

Moreover, we've all heard stories of social engineers dropping USB sticks in the parking lot and waiting for unsuspecting employees to pick them up and plug them in. Yes, that can result in instant compromise of the desktop, depending on what malicious payload is on the USB stick. So another important aspect of protecting endpoints includes defining which devices can connect to a USB port and what those devices can do.

This has been a niche space, but as more disclosure legislation hits around the world, organizations are getting more serious about managing USB ports. As with all other endpoint technologies, device control adds significant management overhead for keeping track of all the mobile devices and USB sticks, and managing the entitlements. The products in this space include management consoles to ease the burden, but managing thousands of *anything* is non-trivial.

Right now device control is a discrete function, but we believe these niche products will also be subsumed into the endpoint suites over the next two years. In the meantime, you may be able to gain some leverage by picking a device control vendor partnered with your endpoint suite provider. Then you should at least be able to centralize the alerts, even if you don't get deeper management integration.

Management Leverage

Though we probably sound like a broken record at this point, keep in mind that each additional security application/capability (control) implemented on the endpoint devices increases the management burden. So when evaluating technology for implementation, be sure to assess the additional management required and the level of integration with your existing endpoint management workflow.

Controls: Full Disk Encryption

It happens quickly. An end user just needed to pick up something at the corner store or a big box retailer. He was in the store for perhaps 15 minutes, but that was plenty of time for a smash and grab. And then your phone rings, a laptop is gone, and it had information on 15,000 customers. You sigh, hang up the phone and call the general counsel — it's disclosure time.

Sound familiar? Maybe this has been you. It likely will be, unless you proactively take action to make sure that the customer data on those mobile devices cannot be accessed by whoever buys the laptop on the gray market. That's right, you need to deploy full disk encryption (FDE) on the devices. Unless you *enjoy* disclosure and meeting with lawyers, that is.

Features

Encryption itself isn't very novel. But managing encryption across an enterprise is, so key management and ease of use end up being the key features that generally drive FDE. As we've harped throughout this series, integration of that management with the rest of the endpoint functions is critical to gaining leverage and managing all the controls implemented on the endpoints.

Of course, that's looking at the issue selfishly from the security professional's perspective. Ultimately the success of the deployment depends on how transparent it is to users. That means it needs to fit in with the authentication techniques they already use to access their laptops. And it needs to just work. Locking a user out of their data, especially an important user at an inopportune time, will make you a pretty unpopular person.

Finally, don't forget about those backups or software updates. If your encryption breaks your backups (and you *are* backing up all those laptops, right?) it's a quick way to find yourself in the unemployment line. Same goes for having to tell the CIO everyone needs to bring their laptops back to the office every Patch Tuesday to get those updates installed.

Integration with Endpoint Suites

Given the natural order of innovation and consolidation, the industry has seen much consolidation of FDE solutions by endpoint vendors. Check Point started the ball rolling by acquiring Pointsec; shortly afterwards Sophos acquired Utimaco and McAfee acquired SafeBoot, so they could each bundle FDE with their endpoint suites.

Now bundling on the purchase order is one thing, but what we are really looking for is bundling from a management standpoint. Can the encryption keys be managed by the endpoint security management console? Is your directory supported natively? Can the FDE policies be set up from the same interface you use for host firewalls and HIPS policies? Unless this level of integration is available, there is little leverage in using FDE from your endpoint vendor.

Free (As in Beer?)

Like all good innovations, the stand-alone companies get acquired and then the capability tends to get integrated into the operating system — which is clearly the case with FDE. Both Microsoft BitLocker and Apple FileVault provide the capability to encrypt at the operating system level (BitLocker is full drive, FileVault is per user). Yes, it's free, but not really. As mentioned above, encryption isn't really novel anymore, it's the *management* of encryption that makes the difference. Neither Microsoft nor Apple currently provides adequate tools to manage FDE across an enterprise.

Which means there will remain a need for third party managed FDE for the foreseeable future, and also that the endpoint security suite is the best place to manage it. We expect further integration of FDE into endpoint security suites, further consolidation of the independent vendors, and ultimately commoditization of the capability.

Building the Endpoint Security Program

Thus far we have looked at what to do right away ([Prioritize](#) and [Triage](#)) and the Controls ([update software and patch](#); [secure configuration](#); [anti-malware](#); [firewall, HIPS and device control](#); and [full disk encryption](#)). But every experienced security professional knows a set of widgets doesn't make a repeatable process, and it's really the process and people that make things secure.

So let's examine how to take these disparate controls and make them into a *program*.

Managing Expectations

The central piece of any security program is making sure you understand what you are committing to and over-communicating your progress. This requires a ton of meetings before, during, and after the process to keep everyone on board. More importantly, the security team needs a standard process for communicating status, surfacing issues, and ensuring there are no surprises in task completion or results.

The old adage about telling them what you are going to do, doing it, and then telling them what you did, is absolutely the right way to handle communications.

Defining Success

The next key aspect of the program is specifying a definition for success. Start with the key drivers for protecting the endpoints. Is it to stop the proliferation of malware? To train users? To protect sensitive data on mobile devices? To improve operational efficiency? If you are going to spend time and money or allocate resources, you need at least one clear driver / justification.

Then use those drivers to define metrics, and operationalize your process based on them. Yes, things like AV update efficiency and percentage of mobile devices encrypted are uninteresting, but you can trend and analyze those metrics. You also can set expectations at the front end of the process about acceptable tolerances for each one.

Think about the relevant incident metrics. How many incidents resulted from malware? User error? Endpoint devices? These numbers have impact, good or bad. And ultimately they are what the senior folks worry about. Operational efficiency is one thing — incidents are another.

These metrics become your dashboard when you are communicating to the muckety-mucks. And remember to use pie charts. We hear CFO-types like pie charts. Okay, I'm kidding.

User Training

Training is the third rail of security, and requires discussion. We are fans of training. But not crappy check-the-box-to-make-the-auditor-happy training. Think more like phishing internal users, and using other social engineering tactics to show employees how exposed they are. Good training is about user engagement. Unfortunately most security awareness training sucks.

Keep the goals in mind. The end user is the first line of defense (and for mobile professionals, unfortunately also the last) so we want to make sure they understand what an attack looks like and what to do if they think they might have a problem. They don't have to develop security kung fu, they just need to understand when they've gotten kicked in the head. For more information and ideas, see [our post on training](#).

Operational Efficiencies

Certainly one of the key ways to justify investment in any kind of program is via operational efficiencies, and in the security business that means automation whenever and wherever possible. So think about the controls we have discussed, and how to automate them. Here's a brief list:

- **Update and Patch, Secure Configurations:** A tool to automate configuration management can kill these two birds with one stone. You set a policy, and it takes care of enforcing standard configurations and keeping key software updated.
- **Anti-malware, FW/HIPS:** With an endpoint suite enforcing policies on updates, software distribution, and policy updates are fairly trivial. Here is the leverage (and the main justification) for consolidating vendors on the endpoint — just beware folks who promise integration but fail to deliver *useful* synergy.
- **Device control, full disk encryption, application white listing:** These technologies are currently less integrated into the endpoint suites, but as the technologies mature, markets consolidate, and vendors actually get out of their own way and integrate the stuff they buy this will get better.

Ultimately, operational efficiencies are all about integrating management of the various technologies used to protect the endpoint.

Feedback Loops

The other key aspect of the program is making sure it adapts to the dynamic nature of the attack space. Here are a few things you should be doing to keep an endpoint program current:

- **Test controls:** We are big fans of hacking yourself, which means using hacking tools to test your own defenses. Check out tools like Metasploit and the commercial offerings, and send phishing emails to your employees trying to get them to visit fake sites — which presumably would pwn them. This is a critical aspect of the security program.
- **Endpoint assessment:** Figure out to what degree your endpoints are vulnerable, usually by scanning devices on connect with a NAC-type product, or with a scanner. Look for patterns to identify faulty configuration, ineffective patching, or other gaps in the endpoint defenses.

- **Configuration updates:** A couple times a year new guidance emerges (from CIS, NIST, etc.) recommending changes to standard builds. Evaluating those changes, and figuring out whether and how the builds should change, helps ensure endpoint protection is always adapted to current attacks.
- **User feedback:** Finally, you need to pay attention to the squeaky wheels in your organization. Perhaps not as much as they demand, but you do have to figure out whether they are complaining about draconian usage policies — and more importantly whether controls are impeding their ability to do their jobs. That’s what we are trying to avoid.

The feedback loops will indicate when it’s time to revisit the controls, perhaps changing standard builds or considering new controls or tools. Keep in mind that without process and failsafes to keep it relevant, all the technology in the world can’t help.

Endpoint Incident Response

Nowadays the endpoint is the path of least resistance for the bad guys to get a foothold in your organization. This means we have to have a structured plan and practiced process for dealing with endpoint compromises. The high level process we'll lay out here focuses on: confirming the attack, containing the damage, and then performing a post-mortem.

To be clear, incident response and forensics are a very specialized disciplines, each with its own host of hairy issues, and best left to the experts. That said, there are things you as a security professional need to understand to avoid interfering with forensics.

Confirming the Attack

There are lots of ways your spidey-sense should start tingling that something is amiss. Maybe it's the user calling up and saying their machine is slow. Maybe it's your SIEM detecting some weird log records. It could be your configuration management system noting some strange executables. Or perhaps your network flow analysis shows reconnaissance activities from the device. A big part of security management is to be able to fire alerts when something suspicious is happening.

Then we make like bloodhounds and investigate the issue. We've got to find the machine and isolate it. Yes, that usually means interrupting the user and *inviting* them to grab a cup of coffee, while you figure out what a mess they've made. The first step is likely to do a scan and compare with your standard builds (you remember the standard build, right?). Basically we look for obvious changes that cause the issues.

If it's not an obvious issue (think tons of pop-ups), then you've got to go deeper. This usually requires forensics tools, including stuff to analyze disks and memory to look for corruption or other compromise. There are lots of good tools (both open source and commercial) available to throw into your forensics toolkit.

We recommend you take a basic course in forensics as you get going for a pretty simple reason. You can really screw up an investigation by doing something wrong, in the wrong order, or using the wrong tools. If it's truly an attack that your organization might want to prosecute at some point, that means you need to maintain *chain of custody* on any evidence you gather. You should consult a forensics expert, and probably your general counsel, to identify the stuff you need to gather for prosecution.

Containing the Damage

"Houston, we have a problem..." Unfortunately your fears were justified, and an endpoint or 200 have been compromised, so now what to do? First off, you should already know what to do, because you have a documented incident response plan and you've practiced the process countless times, and your team springs into action without prompting, **right**? OK, this is the real world, so hopefully you have a plan and your team doesn't look at you like an alien when you take it to DEFCON 4.

In all seriousness, you need an incident response plan. And you need to practice it. The time to figure out your plan stinks is *not* when you have a worm proliferating through your innards at an alarming rate. We aren't going into full depth on that process in this paper, but the general process is as follows:

- **Quarantine:** Bad stuff doesn't spread through osmosis — you need the network to allow malware to find new targets and spread like wildfire, so first isolate the compromised device. Yes, user grumpiness may follow, but whatever. They got pwned, so they can grab a coffee while you figure out how to contain the damage.
- **Assess:** How bad is it? How far has it spread? What are your options to fix it? The next step in the process is to understand what you are dealing with. From confirming the attack you probably have a pretty good idea what's going on. But now you must figure out the best option(s) to fix it.
- **Workaround:** Are there settings that can be deployed on the perimeter or at the network layer that can provide a short term fix for the vulnerability? Maybe you can block communication to the botnet's command and control with an egress filtering rule. Or block inbound traffic on a certain port or some specific non-standard protocol that would stop proliferation of the attack. Obviously be wary of the ripple effect of any workaround (what else does it break?), but allowing folks to get back to work quickly is key, so long as you can avoid the risk of further damage.
- **Remediate:** Is it a matter of changing a setting or uninstalling some bad stuff? That would be optimistic, eh? Now is when you figure out how to fix the issue, and increasingly re-imaging is the best answer. Today's malware hides itself well enough to make it almost impossible to clean a compromised device with certainty. Which means part of your incident response plan should be a leveraged way to re-image machines.

At some point you need to figure out if this is an incident you can handle yourself, or if you need to bring in the big artillery — forensics experts or law enforcement. Part of your IR plan needs to identify the scenarios under which each happens. You don't want something like that to be a judgement call in the heat of battle. So define the scenarios, establish the contacts (within both forensics firms and law enforcement), and be ready. That's what IR is all about.

Post-Mortem

Once folks are done cleaning up an incident, they think the job is done. Well, not so much. The reality is that the job has only just begun — you need to figure out what happened and make sure it doesn't happen again. It's OK to get nailed by something you haven't seen before (fool me once, shame on you). It's not OK to get nailed by the same thing again (fool me twice, shame on me). So you have to schedule a post-mortem.

The post-mortem is not about laying blame, it's about making sure it doesn't happen again. So you need someone to candidly and in great detail understand what happened and where the existing defenses failed. Again, it is what it is and it's the organization must be able to accept failure and move on. But not before you figure out whether process, controls, product, or people need to change.

By the way, it's really hard to fight human nature and build a culture where failure is OK and post-mortems are learning experiences, as opposed to a venue for finger pointing and everyone covering their respective asses. But we don't believe you can be successful at security without a strong incident response plan and that involves a process to unemotionally do a post-mortem.

Endpoint Compliance Reporting

No matter what we do from a security context — whatever the catalyst, budget center, or end goal — we need to substantiate implemented controls. We can grind out teeth and curse the gods all we want, but in today's world, security investments are contingent on some kind of compliance driver.

So we need to focus on documentation and reporting *for everything we do*. Further, we discussed operational efficiencies in the security programs section, and the only way to get any kind of leverage from an endpoint security program is to automate the reporting.

Document What?

First we need to understand **what** needs to be documented from an endpoint perspective for the regulations/standards/guidance you deal with. You must be able to document the process/procedures of your endpoint program, as well as the data substantiating the controls. Process either exists or it doesn't, so that documentation should be straightforward to produce.

On the other hand, figuring out which data types correlate to which controls and apply to which standards requires a big matrix to handle all the permutations and combinations. There are two ways to do this:

- **Buy it:** Many of the IT GRC tools out there help to manage the workflow of a compliance program. The key capability here is the built-in *map*, which connects technical controls to regulations, ostensibly so you don't have to. But these tools cost money and provide limited value.
- **Build it:** The other option involves going through your regulations and figuring out relevant controls. This is about as fun as a root canal, but it has to be done. Most likely you can start with something your buddies, auditor, or vendors have. Vendors have excellent motivation to figure out how their products — representing a variety of security controls — map to the various regulations their customers need to address. The data is out there — you just have to assemble it.

Actually, there is a third option: to just license the content from an organization like the [Unified Compliance Framework](#) folks. They license a big-ass spreadsheet with the map, and their pricing is rather economical.

Packaging

Now that you know what you need to report on, how do you do it? This question is bigger than an endpoint security program — it applies to every security program you run. We recommend you think architecturally. You've got certain domains of controls — network, endpoint, data center, application, etc. You want to put together a few things for each domain to make the auditor happy:

- **Control list:** Go back to your control maps and make a big list of all the controls required for the auditor's checklist (they all have checklists). Make sure the auditor buys into your list, and that you aren't missing anything.
- **Logical architecture:** Show graphically (a picture is worth a thousand words) how your controls are implemented. Every control on the list should appear in the architecture.
- **Data:** You didn't really think the auditor would just believe your architecture diagram, did you? Now you need the data from each of your systems (endpoint suite, configuration management, full disk encryption, etc.) to show that you've actually implemented the controls. Your vendor likely has a pre-built report for your regulation, so you shouldn't have to do a lot of manual report generation.

To be clear, one of the value propositions of IT GRC and other compliance automation products like log management/SIEM is to aggregate all this information (not just from the endpoint program, but from all your programs) and spit out an integrated report. For the most part, with a bit of angst in deployment, these tools can help reduce the burden of preparing for frequent audits across multiple regulations for global enterprises. The question to answer is whether the tool can pay for itself in terms of saved time and effort: is the ROI sufficient?

The answer to that question will be largely dependent on the maturity of the security/risk/compliance organization. As was said of electronic data interchange (EDI) a decade ago, automating a bad process just helps an organization do the wrong things faster. But to the degree that data sources and processes are aligned in a mature fashion, compliance automation tools can improve efficiency and reduce costs.

Dealing with Deficiencies

One other thing to consider is the reality of an audit pointing out a specific deficiency in your endpoint security program. The auditor/assessor is there to find problems, and likely they will. But that doesn't mean the auditor is right.

There, we said it. Sometimes auditors take liberties and/or subjectively decide how to interpret a specific regulation. If there is a specific reason you decided to either bypass a control — or more likely, implement a compensating control — make your case.

In the event (however unlikely) there is a legitimate deficiency, you need to fix it. Welcome, Captain Obvious! During the next audit, first go through the list of previous deficiencies and how you've remediated them. Make a big deal of how you addressed them, which will get the audit off on the right foot.

Who We Are

About the Author

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, like 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.

Mike started as a programmer and a networking consultant, 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 VP Marketing roles at CipherTrust and TruSecure — providing experience in marketing, business development, and channel operations for both product and services companies.

After getting fed up with vendor life, he started Security Incite in 2006 to provide a voice of reason in an overhyped but underwhelming security industry. After taking a short detour as Senior VP, Strategy and CMO at eIQnetworks to chase shiny objects in security and compliance management, Mike joins 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](#) 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, L.L.C.](#) 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:

- *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 follow 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 assessments.
- *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, merger and acquisition assessments, and more. 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 services tend to be customized to meet a client's specific 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.