Implementing and Managing Patch and Configuration Management

Version 1.2

Released: November 29, 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 Lumension Security, Inc.

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 Texas, Florida, Washington D.C., Ireland, Luxembourg, Singapore, the United Kingdom, and Australia. Lumension: IT Secured. Success Optimized.™ More information can be found at www.lumension.com.

Copyright

Table of Contents

Introduction

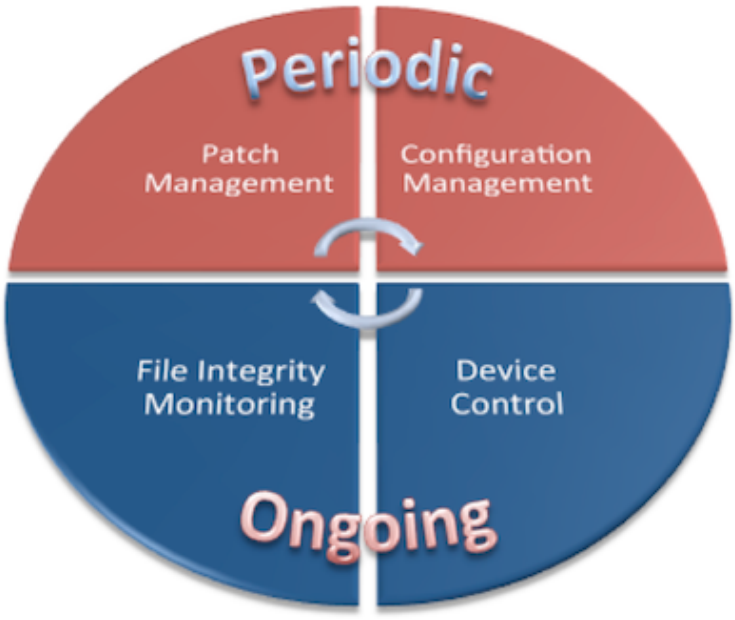
Endpoint devices have been the bane of security practitioners for as long as we can remember. Whether it's unknowing users who click anything, folks who don't think the rules apply to them, or the forgetful sorts who just leave their devices anywhere and everywhere, maintaining control over endpoints causes heartburn at many organizations. To address these concerns, Securosis recently published an Endpoint Security Management Buyer's Guide, which began with a list of the key issues complicating endpoint security management, including:

- **Emerging Attack Vectors:** Everyone wants to talk about advanced attacks because they are exciting and sexy, but many successful attacks stem from simple operational failures. Whether it's an inability to patch in a timely fashion, or to maintain secure configurations, far too many people leave the proverbial barn doors open on their devices. Or attackers target users via sleight-of-hand and social engineering. Employees unknowingly open the doors for attackers… and enable compromise. That doesn't mean you don't have to worry about advanced malware or persistent attackers, but if your operational house isn't in order yet, don't put the cart before the horse.

Everyone wants to talk about advanced attacks because they are exciting and sexy, but many successful attacks stem from simple operational failures.

- **Device Sprawl:** A typical organization has a variety of PC variants running numerous operating systems. They may be virtualized and may connect from anywhere in the world – including networks you do not control. Even better, many employees carry smartphones in their pockets and tablets in their backpacks, but those are just more computers with access to your critical data. Any endpoint security management controls and processes need to be enforced consistently across the sprawl of all your devices.
- **BYOD:** Corporate mobile devices are just the tip of the iceberg – organizations are increasingly supporting BYOD (Bring Your Own Device) policies, which require you to protect not only corporate assets but employees' personal devices as well. So you may need to support any variety of PC, Mac, smartphone, or tablet any employee wants to use. This requires the ability to granularly manage device policies. For extra fun, patching an app on an employee device could break a capability which its *owner* relies on.

To provide this more strategic view of endpoint security management, we identified 4 specific controls typically used to manage the security of endpoints, and broke them up into periodic and ongoing controls, depicted below.



Here is a quick refresher on patch and configuration management:

- **Patch Management:** Patch managers install fixes from software vendors to address vulnerabilities. The best known patching process is from Microsoft on a monthly schedule. On Patch Tuesday, Microsoft issues a variety of software fixes to address defects that could result in exploitation of their systems. Once a patch is issued your organization needs to assess it, figure out which devices need to be patched, and install relevant patches within the window specified by policy – typically a few days. A patch management product scans devices, installs patches, and reports on the success and failure of the process.

- **Configuration Management:** Configuration management enables an organization to define an authorized set of configurations for devices in use within the environment. These configurations govern the applications installed, device settings, services running, and security controls in place. This is important because a changing configuration might indicate malware manipulation or an operational error. Additionally, configuration management can help ease the provisioning burden of setting up and reimaging devices. Configuration management enables your organization to define what *should* be running on each device based on entitlements, and to identify non-compliant devices.

You bought the technology – what now? It's time to implement and manage your new toys, so this paper will provide a series of processes and practices for successfully implementing and managing patch and configuration management tools. You need to start by deciding whether you want a Big Bang type deployment, or whether an incremental, structured rollout plan makes more sense. Let's dig into these two deployment models and how they affect implementation and ongoing management.

Quick Wins for long term success

One of the main challenges in implementing any security technology is to show immediate value to justify the investment. Of course you can install patches and manage configurations manually, or using built-in and/or free utilities. When spending money on patch and configuration management you need to focus on value – above and beyond what you already had – so we will break the implementation process into two phases, described below:

- The *Quick Wins* process is for initial deployments. Its focus is on rapid deployment on critical devices with access to sensitive data. You then take the opportunity to fine-tune deployment and policies, which streamlines the path to full deployment later.

- The *Full Deployment* process is for the long haul. It is a methodical series of steps to full enforcement of enterprise patch and/or configuration policies. The goal of both controls is to minimize exposure, which means ensuring patches are applied as quickly as practical and monitoring configurations to ensure malware hasn't made unauthorized configuration changes.

The key difference is that the Quick Wins process doesn't cover every endpoint – just the most important ones. It gets you up and running quickly, and sets the stage for full deployment.

The key difference is that the Quick Wins process doesn't cover every endpoint – just the most important ones. It gets you up and running quickly, and sets the stage for full deployment. Full Deployment is where you dig in, spend more time, and implement long-term policies across all devices. Full coverage is critical because today's attackers often do not go directly after sensitive data stores. They tend to start slowly, gaining presence via known vulnerabilities and configuration mistakes, patiently moving laterally through the environment until they reach their target.

We designed these processes to complement each other. If you start with Quick Wins, all your work feeds directly into Full Deployment. If you already know where you want to focus and have a mature endpoint management infrastructure, you can jump right into Full Deployment. Either way, our process guides you around common problems and should help speed implementation.

Getting started

No matter whether you choose Quick Wins or Full Deployment, we break the implementation process into four major steps:

1. **Prepare:** Determine which model you will use, define priorities among users and devices, and build consensus on the processes to be used. You will also need to ensure all parties involved understand their roles and will accept responsibility for results – including not only security scanning and monitoring functions, but also the operations folks in charge of remediating any issues.

2. **Integrate and Deploy Technology:** Next you will determine your deployment architecture and integrate with your existing infrastructure. We cover most integration options – even if you only plan on a limited deployment (and no, you don't have to do everything at once). This involves not just setting up the endpoint security management platform, but also deploying any required agents to manage devices.

3. **Configure and Deploy Policies:** Once the pieces are integrated you can configure initial settings and start policy deployment. Patch and configuration management policies are fundamentally different, so we will address them separately.

4. **Ongoing Management:** At this point you should be up and running. Managing is all about handling incidents, deploying new policies, tuning and removing old ones, and system maintenance.

This paper will go into each step in depth, focusing on what you need to know to get the job done. Implementing and managing patch and configuration management doesn't need to be intimidating, so we focus on what you need to know to make progress with quick value, within a sustainable process.

Preparation

You know the old saying, "if you fail to prepare, you prepare to fail." It's true, and the pre-deployment preparation for patch and configuration management involves ensuring your processes are solid, defining device coverage and roll-out priorities, figuring out what's already out there, and a testing phase to make sure you are ready for broad deployment. So let's examine the patch and configuration management processes.

Determine Processes

We are process centric at Securosis. We have a deep appreciation for the folly of trying to implement and manage technology without proper processes and defined accountabilities *before* products get installed. So we start most activities with a check to ensure the process supports the problem to be solved. With patch and configuration management there are two distinct but tightly intertwined processes.

We have a deep appreciation for the folly of trying to implement and manage technology without proper processes and defined accountabilities *before* products get installed.

Of course you don't need all the functions below. Figure out which steps will work for your organization. But you do need to make sure everyone understands what they are supposed to do – especially when it comes to remediation. If the operations team is expected to run through the patch process, open up maintenance windows, and confirm the successful implementation of each patch, they need to know. Likewise, if the incident response team needs to investigate strange configuration changes found during assessment, the handoffs must be clearly defined — along with your ability to remediate a device under investigation.

Patch Management

1. **Discover and define targets:** Before you jump into the patch management process, you need to decide which devices will be included. Is it just endpoints, or do you also need to patch servers? These days you also need to consider cloud instances. The technology is largely the same, but increased quantities of devices make execution more challenging.

2. **Obtain patches:** You need to monitor for release of relevant patches, and then figure out whether you need each patch, or you can work around the issue.

3. **Prepare to patch:** Once each patch is obtained you need to figure out how critical the issue is. Is it something you need to fix right now? Can it wait for the next maintenance window? Once priority is established, give the patch a final Q/A check to ensure it won't break anything important.

4. **Deploy the patch:** Once preparation is complete and your window has arrived, you can install.

5. **Confirm the patch:** Patches don't help if the install fails, so confirm that each patch is fully installed.

6. **Reporting:** Compliance requirements for timely patching make reporting integral.

Obviously this is a very high-level process. If you need a much more granular process map for patch management, with metrics and cost models, check out Patch Management Quant.

Configuration Management

1. **Establish configuration baselines and/or benchmarks:** First define acceptable secure configurations for each managed device type. Many organizations start with benchmarks from CIS, NIST, or their endpoint security management vendors for granular guidance on how devices should be configured.

2. **Discover and define targets:** Next find the devices that need to be managed. Ideally you would leverage an endpoint security management platform with an integrated asset management repository. You will also want to categorize and group assets to avoid unnecessary services. Engineering workstations, for example, require different configurations than Finance systems.

3. **Assess, alert, and report changes:** Once devices are discovered and categorized, define a frequency for assessments. How often will you check them against policy? Vendors talk about "continuous assessment" but assessments aren't really continuous. Fortunately this isn't normally a problem – not least because most operational groups wouldn't be able to validate alerts and correct issues in real time anyway.

4. **Remediate:** Once a problem is identified, either it needs to be fixed or someone needs to grant an exception. You are likely to have too much work to handle it all immediately so prioritization is key. We offered some guidance on prioritization for vulnerability management, but the concepts are the same for configuration management. You will also probably need to verify that changes actually took place for the audit, as well as plan for rollback in case the change breaks something.

Define Initial Priorities/Targets and Deployment Model

After gaining consensus on the applicable processes and ensuring everyone knows their roles and responsibilities, it's time to determine your initial priorities and targets to figure out whether you will start with the Quick Wins process or jump right into Full Deployment. Most organizations have at least a vague sense of what types of devices they need to patch and manage, but translating that into deployment priorities can be tricky. Let's highlight some of the categories of things you can manage to help figure out the best direction.

- **Servers:** Keeping server devices (more specifically their operating systems) updated is essential for protecting them. Look to group servers logically based on function, so you can identify typical configurations and applicable patch windows for each class of device. Also factor in whether you are dealing with physical servers, private cloud instances, or public cloud instances — because managing each type differs dramatically.

- **PCs:** Non-server PCs are rarely the ultimate target, but they provide a way for attackers to gain a foothold within your organization so they can jump laterally to attack servers. Group PCs logically based on job function and need for access to critical data stores. Keep in mind that laptops create unique problems for patch and configuration management because they may connect to the network infrequently, so consider whether you want to tackle that as part of the initial deployment.

- **Mobile devices:** Quicker than you can say BYOD, you will need to effectively manage mobile devices (including smartphones) that access your network. Smartphone vendors provide utilities to update and enforce configuration policies on their devices, but in heterogeneous environments it is useful to provide consistent patch and configuration management across all devices. This may be constrained by organizational structure — particularly if a different group is responsible for mobile devices and pushes for its own purpose-built tool.

- **Applications:** Finally, applications have emerged as the path of least resistance for many attacks, so keeping commercial applications patched and properly configured has become even more important. The good news is that patch and configuration management platforms handle applications and operating systems similarly, so supporting applications is not likely to require massive technology changes, as long as you are using an application that has integrated with the leading patch tools. But you will need to decide whether application patching is something to deal with in your initial deployment.

The next question is which deployment process to use. The easy answer is almost always the same: *start with the Quick Wins process.*

You should now have a sense of what devices to focus on and where to start. The next question is which deployment process to use. The easy answer is almost always the same: *start with the Quick Wins process.* The only common exception is when you have already prioritized what to manage, have a good sense of where you need to manage it, and believe you understand the scope you need to tackle – then you might be able to jump directly to Full Deployment. This tends to come up when you have a specific compliance deficiency or have tracked back a breach to poor device hygiene, because going all in on patch and configuration management will solve that problem. Otherwise we suggest starting with Quick Wins to highlight what works and doesn't, and to help figure out where to focus your full deployment.

Initial discovery

Both deployment processes start with a discovery phase to figure out what's actually out there. We suggest you kickstart the effort by mining an existing asset management repository – perhaps a CMDB, enterprise directory, or other device data store. Of course your asset management function can only provide detail on devices you already know about, so it is important to perform an active scan of applicable IP address ranges to discover what's really out there. You are likely to be surprised when you compare reports against reality.

Do yourself a favor and be sure to coordinate with the operations team to avoid disrupting production systems. And note that a segmented network architecture (to segregate servers within PCI scope, for instance) will require access to the protected segments for scanning. If you can scan your entire environment from a single location without consulting the operations team, you have bigger problems than patch management, so maybe you need to address those first.

Test and Proof of Concept

We talked a bit about this in the Endpoint Security Management Buyer's Guide, because a Proof of Concept (PoC) is typically part of the procurement process one way or another. But let's distinguish between the kind of testing you do before you buy and the testing you need before you implement. During the selection process you focus on user experience, deployment architecture, device & application coverage, and the quality and detail of reports – keeping in mind your long-term endpoint security management needs.

Pre-deployment testing is all about figuring out what breaks when you implement. Don't decide to patch all your Windows devices and *then* find out that XP isn't supported as well as it should be. Or that patching Adobe Reader is unreliable. Make sure you don't create more problems than you solve. You need to make sure your initial set of supported platforms and apps patch reliably and save time.

Make sure you don't create more problems than you solve. You need to make sure your initial set of supported platforms and apps patch reliably and save time.

As part of this limited deployment, look for a representative sample of the devices you decided to support in the implementation pilot. Pick this sample from folks who won't react too poorly if things don't work perfectly. IT staffers are usually good guinea pigs, as are more technically sophisticated employees.

The testing process also provides a good mechanism to train staff on the tools – especially whoever wasn't involved in the proof of concept. Remember, they will have to both run the tool and remediate the issues it uncovers. So having them run through the process during testing will pay dividends when live ammo is flying.

Integrate and Deploy Technologies

By this point planning should be complete. You have designed your patch and configuration management processes, defined priorities to manage the devices in your environment, figured out which high-level implementation process to start with, discovered the devices in your environment, and performed initial testing to make sure the new technology doesn't break anything. Now it's time to integrate the patch and configuration management tools into your environment. Enough of this planning stuff — let's get down to business! But you won't actually remediate anything yet – the initial focus is on integrating technical components, installing agents as necessary, and preparing to flip the switch.

Component Overview

We are grouping patch and configuration management together, so we will talk about common concepts such as management servers and agents. A management server might be specifically associated with a patch management product and/or the configuration management environment, or both. Obviously leverage between the two is useful, but depending on which technologies you selected to deploy as part of your endpoint management security environment, you might have different consoles and agents. But the deployment considerations are similar, regardless of the specifics.

If an attacker can change the configuration of a device or apply a malicious patch, it's pretty much game over. So it's important to make sure the components are deployed correctly with appropriate security

Before we describe specific components we need to briefly go over the inherent security requirements of the different components. If an attacker can change the configuration of a device or apply a malicious patch, it's pretty much game over. So it's important to make sure the components are deployed correctly with appropriate security controls. Most solutions use some type of cryptography, both for authentication and to protect communication between components. We are not religious about specific authentication mechanisms (PKI or Windows or whatever), but be sure to check for recent attacks or vulnerabilities in whichever technologies you depend on. You may also want to consider two-factor authentication or some kind of privileged user management to better protect the management console.

You will also need to coordinate with the network team to ensure the proper firewall ports are open (and/or proxies identified) to receive updates and new patches from vendors, and to communicate with the relays and/or endpoints using the ports specified by your endpoint security management vendor. Be considerate of the network security team, of course, who will likely resist opening up all sorts of ports throughout the environment. Default deny is still your friend – so when planning the deployment make sure you understand where the servers, distribution points & relays, and agents will be, and how they communicate.

Management Server/Appliance

The management server is the brains of the operation. It holds the policies and serves as the focal point for data aggregation, analysis, visualization, and reporting. You have a few options for how to implement the management server, with pros and cons to each.

- **Software:** The most common choice is to install software on a dedicated server. Depending on your product this might actually run across multiple physical servers for different internal components such as a back-end database, or to distribute functions for better performance. Some products require different software components running concurrently to manage different functions. This is frequently a legacy of mergers and acquisitions – most products converge on a single software base over time, although integration may not be as complete as you would expect. Management server overhead is generally fairly low, especially outside large enterprises, so this server often handles some network monitoring, functions as an email MTA for alerting, and manages endpoint agents. A small-to-medium-sized organization generally only needs to deploy additional servers for load balancing or hot standby. Integration is easy – install the software and position the physical server as needed, based on deployment priorities and network configuration, ensuring visibility to the relays and/or agents that need to communicate with it.

- **Appliance:** In this scenario the endpoint security management software comes preinstalled on dedicated hardware, presumably with a locked-down secure operating system. There is no software to install, so the initial integration is usually a matter of connecting it to the network and setting a few basic options – we will cover the full configuration later. As with a standard server, the appliance usually includes the ability to run multiple functions, but you might need licenses to unlock capabilities.

- **Virtual Appliance:** The endpoint security management software is preinstalled into a virtual machine for deployment as a virtual server. This is similar to an appliance but requires work to get up and running on your virtualization platform of choice, configure the network, and then set up the initial configuration options as if it were a physical server or appliance.

For now just get the tool up and running so you can integrate the other components. Do not deploy any policies or turn on monitoring yet.

Agents

Endpoint agents are, by far, the most varied patch and configuration management components. Fortunately, as widely as features and functions vary, the deployment process is consistent.

1. **Test, then test more:** We know we keep telling you to test your agents before production use, because inadequate testing is the single most common problem people encounter. If you haven't already, be sure to test your agents on a variety of real-world systems in your environment to make sure performance and compatibility are acceptable. That's why choosing test devices is so important.

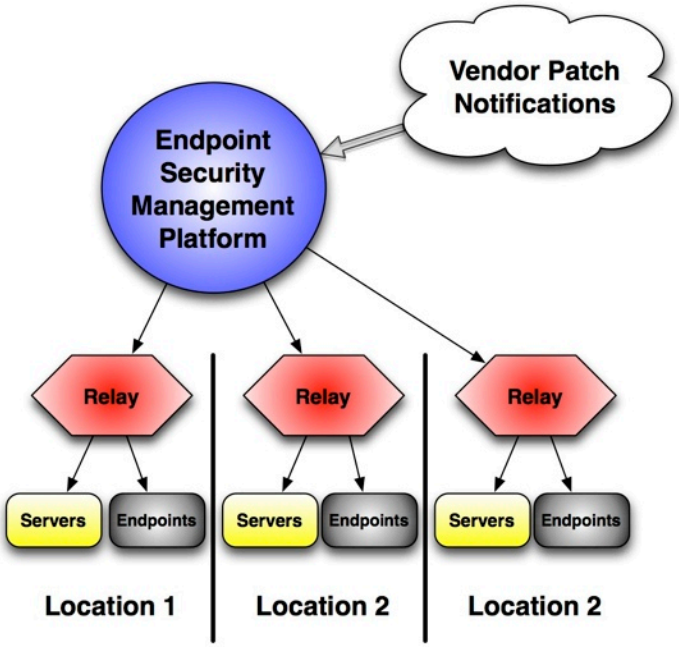
2. **Create a deployment package or enable in your EPP tool:** The best way to deploy any agent is to use whatever software distribution tool you already use for normal system updates. There is no need to reinvent the wheel. This means building a deployment package with the agent configured to connect to the patch and/or configuration management server. Remember to account for any network restrictions that could isolate endpoints from the server. In some cases the agent may be integrated into your existing EPP (Endpoint Protection Platform) tool. More often you will need to deploy an additional agent, but if it is fully integrated you can configure and enable it either through the patch/configuration management console or in the EPP tool itself.

3. **Activate and confirm agent installation:** Once the agent is deployed go back to your patch/configuration management console to validate that systems are covered, agents are running, and they can communicate with the server. Don't turn on any policies yet – for now just confirm that the agents deployed successfully and are communicating.

Some offerings do not require agents *per se,* but let's be clear that software or a remote access capability is required on *any* managed device to check configurations and perform any required remediation. It could be a 'dissolvable' agent, which is downloaded as necessary by the endpoint and deleted when it's no longer needed. Agentless options can work for patch management because scans and patching are both performed on a periodic basis. Configuration management is a bit different – monitoring configuration changes requires fairly frequent assessment, more likely hourly than monthly – so keep that in mind when determining whether to deploy agents. We aren't hung up on whether or not to use agents – just understand the trade-offs when deciding.

Deployment Models

Making sure the patch and/or configuration management system will scale is critical. Scaling many vendor architectures involves distribution points (also called relays) to aggregate patches, analyze, scan, and normalize and compress aggregated data to be sent back to the central console. Some of the patches to deploy will be large (gigabytes), so you don't want every device contacting your master server for each download.



You are likely to have a central server as the main contact point with the vendor's information service, to receive notification of new patches and download packages. Then you will have a bunch of distribution points (relays) by location, bandwidth, and numbers of managed devices. They are typically configured as slaves of the central server, and receive policies from it.

You will also want to consider high availability architectures for larger environments. They require configuring management servers, and perhaps relays, with hot standbys to handle failover. Pay attention to the server replication mechanism – you shouldn't lose data if you lose a server. Here are some other considerations when implementing the technology:

- Pay attention to component security, including communications channels and protocols. Hopefully you can piggyback on the existing VPN between your locations, with additional network-layer security between components, including mutual authentication.
- If you plan to allow remote locations to implement their own policies for patch and/or configuration management, now is the time to set up a few test policies and a workflow to verify that your tool can support your requirements. Make sure to factor in some kind of policy authorization process to ensure each location adheres to the organization's general policies.

Remote Devices

One of the fun parts of managing thousands of endpoints is that at any time quite a few are not connected to your network. But they still have access to sensitive data, and require enforcement of patch and configuration policies. There are two considerations when thinking about these remote devices:

1. **Agent Implementation:** Lack of full-time network access to devices raises the challenge of how to get the implemented onto the devices in the first place. So you will need a process to access every single

device and install the agent. We discussed endpoint protection platforms above, and those vendors have solved this problem, so this is another case where piggybacking on EPP infrastructure can facilitate things — some vendors have open platforms which enable third-party vendors to leverage management functions. Otherwise you might need to connect remotely into the devices or utilize something like a GPO policy to run a script once the device connects to the network (for instance, when they pick up email or access a file store).

2. **Assessment and Remediation:** Once the agent is in place you need to ensure the device establishes a connection every so often so the device can be inspected and remediated as necessary. The agent should be able to phone home as needed, but it's your responsibility to define how often and from where. You will have grumpy end users if you download multi-gigabyte patches over satellite link in the Amazon.

Fortunately software distribution to remote devices is a solved problem, so there is no need to overthink this – just ensure you have suitable policies and infrastructure in place for remote devices.

Other Integration Points

Your patch and configuration management system will need to integrate with a number of other enterprise systems. The first is the asset repository we mentioned earlier. Bi-directional communication with the central asset management environment is important, for both initial discovery and tracking of new devices. If your organization hasn't deployed any kind of centralized asset management, many patch and configuration management platforms include an integrated capability that can serve as the authoritative source for your shop.

Once a patch and/or configuration change is identified, someone needs to do the work. That may mean tasking the operations team with making a change or installing a patch. Most of those teams live and die by their help desk/trouble ticketing systems, so make sure you can both create new tickets and close the loop when a ticket is marked completed.

Finally, there is tremendous value in sending patch and configuration information to a security monitoring/SIEM system. Configuration changes can indicate malware or other attacks, and when correlated with other enterprise network and server data sources, may shorten the detection window for an attack. Likewise, a missing patch on a particular device, combined with information about a specific attack detected by an IPS, can show a clear and present danger to a specific device to trigger action. So ensure your patch/configuration management console can communicate with the SIEM.

Training

The last consideration to mention for integration and deployment is training. Some staff probably received some training during the testing phase, so now is the time to get everyone else on the operational and security teams educated on how to use the tools, as well as their respective responsibilities and accountabilities. Make sure that you are prepared to flip the switch, and can take advantage of the new management systems without causing trouble elsewhere.

Defining Policies

With the pieces in place it is time to configure and deploy policies to prepare for the inevitable patch cycles, and to start monitoring configurations on your key devices. Before we get deep into staging your deployment, keep in mind that we break things out with extreme granularity to fit the full range of organizations. Many of you won't need this much depth, due to organizational size or the nature of your policies and priorities. Don't get hung up on our multi-step process – many of you won't need to move this cautiously and can run through multiple steps quickly.

The key to success is to think incrementally – too often we hear about organizations which *can* pump out a bunch of agents quickly, so they think they should. Endpoints can be finicky devices, and you should be sure to leave adequate time for testing and burn-in before you go all-in. So it's prudent to pick a single device type or group of users, create the appropriate policy, slowly roll out, and tune iteratively until you attain full coverage. We are not opposed to deploying quickly, but we have a keen appreciation for the challenges of fast deployment – especially in managing expectations. Better to under-promise and over-deliver than vice-versa, right?

The key to success is to think incrementally – too often we hear about organizations which *can* pump out a bunch of agents quickly, so they think they should.

So here is a reasonable deployment plan:

1. **Define the policy:** Set policies based on the type of device and what you are doing on it – patch or configuration management. We will dig into the specific policy decisions to make later. Again, we suggest you start with a single device type – possibly even a specific group of users – and expand incrementally once the initial deployment is complete. This helps reduce management overhead and enables you to tune the policy. In most cases your vendor will provide prebuilt policies and categories to jumpstart your own policy development. It's entirely appropriate to start with one of those and refine based on initial results.

2. **Deploy to a subset:** The next step is to deploy the policy to a limited subset (either device types, groups of users, or both) of your overall coverage goal. This limits the number of deployment failures and gives you time to adjust and tune the policy. The key is to start small so you don't get overloaded during the tuning process. It is much easier to grow a small deployment than to deal with overwhelming fallout from a poorly tuned policy.

3. **Analyze and tune:** Iteratively observe results and adjust the policy. If you see too many deployment or remediation failures, or false positives, adjust the policy.

4. **Expand scope:** Once the policy is tuned you can start thinking about expanding your deployment scope and size. You can add additional devices and groups of users, expand the number of applications being patched, etc. Full deployments should rarely happen as a 'big bang', so grow slowly and surely to ensure you don't risk deployment failure by going too far too fast. Smaller organizations can often move quickly to full deployment, but we strongly suggest starting small – even if it's only for a day.

Patch Management Policies

In a perfect world, the patch management engine would just run and you could get back to World of Warcraft. Alas, the world isn't perfect and patch management isn't nearly as automated as we would all prefer. You can automate some aspects of the process (including monitoring for new patches) but ultimately you need to decide which patches get applied, in what order, and build the installation packages. The good news is that once this is done the tools generally do a good job of automating installation, confirmation, and tracking. But there is still significant work up front.

The good news is that once this is done the tools generally do a good job of automating installation, confirmation, and tracking.

Put another way, patch management policies are unique for every patch cycle. Of course you can define consistent aspects of the process (such as maintenance windows and user notifications) for every cycle, but each time you need to decide what to patch and what to skip.

Discovery and Target Definition

Depending on whether you are rolling out a Quick Wins limited deployment, extending an existing deployment, or going all-in with a big bang full deployment, the first step is to load up the system with the devices to be managed. Additionally, you need to decide what to do when a new device is found out of compliance with policy. Do you force a patch deployment right away? You also need to define the frequency of revisiting the asset list (daily, weekly, monthly, etc.), because new devices need some endpoint security management love as well.

Obtain Patches

The next step in patch management is finding the patches applicable to your environment. Here you define your information sources (patch management vendors, operating system and application developers, etc.); then build a process to evaluate what has been patched, and more importantly criteria for what to patch and when. You need to make sure the operations team buys into this criteria because they will need to live with it, and that you have a process for out-of-cycle patches – typically for high-risk 0-day vulnerabilities. You will also test patches to make sure they don't cause more harm than good. Again, there isn't much automation in a patch management tool – it's a process to work through each cycle.

Preparation

Next prepare to deploy the patch — which includes downloading, determining the order of installation, building the distribution package, and getting the software to the relays in preparation to flip the switch. In this step you determine such things as:

- The devices that need to be patched and with what priority
- The criticality of each patch, and its timeframe for deployment
- Whether to force a reboot after deployment
- Alerting levels (if a patch fails to deploy, etc.)
- Notification levels (do you tell the user the patch is being installed?)

Deployment

At this point it's mostly a question of pushing the button and waiting for the magic to happen. The policies are more about whether to roll back in case the patch fails or breaks something, and how to handle exceptions when a patch fails to install. We will talk about reporting later but it comes into play here as well.

Confirmation

Finally, you need to confirm the patch was completely installed and is now operational. This involves another scan of the device in question, with some reporting to substantiate that the patch was installed within the agreed-upon window of time.

Configuration Management Policies

Like to patch policies, configuration management policies are inextricably linked to the process you implemented for configuration management. The good news is that much more configuration management can be automated because the configurations for each device type shouldn't vary much from day to day or even month to month. If proper care is taken in setting up the baselines you shouldn't need to babysit policies much.

Establish configuration baselines and/or benchmarks

As we described earlier, the initial configuration management decisions involve locking in the configuration baselines you will use for each device type. A number of resources are available for kickstarting your efforts, including the CIS benchmarks and NIST guidance (PDF), and each vendor has their own ideas on how to configure each device type. Regardless of where you get your baselines, you need consensus – a large part of your operational job will be to manage the inevitable exceptions when someone needs to have their device configured differently because they are *special*. Always keep the tradeoffs in mind. The more inclusive your baseline, the fewer exceptions you will have to deal with, but the more chances you have to make a poor security choice.

You can try a few different policy constructs when developing baselines. You can set a gold standard for your entire organization, then deal with the folks who need something different. Or you could set different baselines for different constituencies. You know, where executives can do whatever they want at one end of

the spectrum while call center desktops get draconian VDI system images. Finally, you could set the baseline based on device type (PC vs. Mac vs. mobile, etc.). Most likely you will end up mixing and matching between these because one size rarely fits all.

Discover and define targets

As with patch management, you need to load the system up with devices to be managed. You also need to decide what to do when a device is found out of compliance with your policy. Do you force immediate remediation? You also need to define the frequency of revisiting the asset list (daily, weekly, monthly, etc.) likely adopting the same schedule as patch management especially if leveraging an endpoint security management platform for both functions. This ensures all new devices have their configurations monitored.

Assess, alert, and report changes

Once you know what will be managed, move on to defining the finer points of configuration assessment, such as:

- How often will you evaluate the configuration? Will that vary between devices always on the network and remote devices that connect intermittently?
- Which device types need agents, and which can be managed through remote assessment? Does that vary for remote devices or bandwidth-constrained networks?
- How critical are specific configuration changes? For example, does the appearance of a web server on a device require immediate incident response?
- Who needs to be alerted when a device violates policy? Does that vary by device type or user?
- Do certain types of configuration changes get pumped directly into the trouble ticketing system for operational remediation?

Remediate and Confirm

Next you will actually fix the configuration issues found during assessment. There aren't many policy decisions here, aside from how quickly to confirm each change once it is marked completed in the trouble ticketing system (assuming Operations handles the changes). We mentioned integration with the trouble ticket system – you need to close the loop with the ops team to ensure changes are both authorized and completed correctly.

Reporting

The driver behind endpoint security management is often compliance. That means you need to produce artifacts on (document) what the patch and configuration management systems have done for in-scope devices. You are likely to kill a few trees on reports showing progress, demonstrating value, and communicating with other stakeholders. Here are a few ideas for starter reports:

- Compliance reports are a no-brainer and included in every endpoint security management product. For example, showing you scanned all endpoints or servers every month, installed missing patches, and ensured all devices were in compliance with baseline standards will make every security assessment faster.

The driver behind endpoint security management is often compliance. That means you need to produce artifacts on (document) what the patch and configuration management systems have done for in-scope devices.

- User-based reports can highlight which users are model citizens and which aren't. You know, those users who don't patch their devices, ever (until finally you need to reimage) or have ongoing configuration problems from consistent malware infections. You can't make these users follow policy but you can call them out to management when they don't. You can also monitor these devices closely to make sure you detect issues early. This kind of report can also be useful for identifying likely candidates to revisit security awareness education.
- Application and vendor reports can give you a feel for which vendors issue more patches (and cost you more in operations), which could be interesting when it's time to pay for maintenance on those applications and operating systems.
- Trend reports are extremely valuable for showing the value of the tool and how teams are performing operationally. Show patch coverage, devices in compliance with configuration baselines, mean time to patch (especially for out-of-cycle/critical patches), time to address critical configuration problems, etc. Most organizations which generate these reports achieve large improvements over time, in terms of managing endpoints and being more responsive to threats. Never underestimate the political value of a good report showing trends with colorful graphs.

Your typical endpoint security management platform will ship with hundreds of canned reports, many with little or no value. So ensure you can customize the reports quickly and easily to show the information you need, as you need to show it.

Patch Management Operations

As we discussed above, there isn't a huge amount of monthly leverage in operating your patch management process. You need to do the work of monitoring for new patches, assessing each new patch for deployment, testing patches prior to deployment, bundling installation packages, and then installing patches on affected devices. You will perform all these activities each month, whether you enjoy them or not. We have already explained those monthly activities within the context of defining policies, so let's go a step deeper.

Troubleshooting

The biggest patch management issue is patches that fail to install properly, for whatever reason. So the first operational task is to ensure the integrity of the process – that the patch was installed and operates properly. As we described in detail in Patch Management Quant, once the patch is confirmed the tool also needs to clean up any patch residue (temporary files, etc.).

In the event the patch doesn't deploy properly, you go to a clean up step – which involves identifying the failed deployment, determining the reason for the failure, adjusting the deployment parameters, and eventually reinstalling. For instance, here are three typical patch failure reasons which can be isolated fairly easily:

The point is to address one-off situations as necessary, and to make sure there isn't a systemic problem with your process.

1. **Relay fail:** If you deploy a hierarchical environment to better utilize bandwidth, your relay points (distribution servers) might not be operating properly. It could be a server failure or a network issue. If an entire site or location doesn't successfully patch, that's a strong indication of a distribution problem. It's not brain surgery to diagnose many of these issues.
2. **Agent fail:** Another likely culprit is failure of an endpoint agent to do its job. If installation failures appear more random this might be the culprit. You will need to analyze the devices to make sure there are no conflicts and that users didn't turn off or uninstall the agent.
3. **Policy fail:** As unlikely as it is, you (or your ops folks) might have configured the policies incorrectly. This is reasonably common – you need to set up policies each patch cycle, and nobody is perfect.

There are many other reasons a patch might not deploy properly. The point is to address one-off situations as necessary, and to make sure there isn't a systemic problem with your process. You will use this kind of troubleshooting analysis and data to move on to the next step of operating your patch environment: optimization.

Optimization

Just like any other optimization process, this one starts with a critical review of the current operation. What works? What doesn't? How long does it take you to patch 75% of your managed devices? 90%? 100%? Is that increasing over time, or decreasing? What types of patches are failing (operating systems, applications, servers, endpoints, or something else)? How does device location (remote vs. on-network) affect success rates? Are certain business units more successful than others? During the review, consider adding new policies and groups. But be careful — patch management requires a substantial manual effort each month, so there is a point of diminishing returns from very rigid policies intended to achieve better automation.

If you find the environment reasonably stable, periodic reviews become more about tuning polices than overhauling them. This involves revisiting your deployment and figuring out whether you have the right hierarchy to effectively distribute patches. Do you need more distribution points or less? Are you optimizing bandwidth? Do you need to install agents to achieve more granular management? Or perhaps remove agents if you can patch without them.

Look for incremental improvement — changes should be thoroughly planned out and structured.

Look for incremental improvement — changes should be thoroughly planned out and structured. This enables you to isolate the effect of each change and reevaluate each aspect iteratively. If you change too much at one time it will be difficult to figure out what worked and what didn't.

Pay attention to maintenance of your environment. The servers and distribution points need to be backed up and kept current, along with updating agents as needed. Obviously you need to test infrastructure software updates – just like any other patch or update – prior to deployment, but the patching system itself could be an attacker's target, so you need to keep it up to date as well. We tend to be wary of automatic updating for most enterprise security tools – there are too many example of bad updates wreaking havoc. Improvements in quicker implementation can easily be lost if you take down your environment while you try to back out a busted patch.

Documentation

Finally, you defined a bunch of reports earlier in the process, to run on an ongoing basis. Obviously you need these artifacts for compliance purposes, but pay attention to the operational data they generate yourself. Feed that information back into the process to continually improve patch management.

Configuration Management Operations

The key high-level difference between configuration and patch management operations is that configuration management offers more opportunity for automation. Unless you are changing standard builds and/or reevaluating benchmarks – then operations become more of a high-profile monitoring function. You will be alerted to a configuration change, and like any other potential incident you need to investigate and determine the proper remediation as part of a structured response process.

Continuous Monitoring

The first operational question is frequency of assessment. In a perfect world you would like to continuously assess your devices, to shorten the window between attack-related configuration change and detection. Of course there is a point of diminishing returns, in terms of device resources and network bandwidth devoted to continuous assessment. Don't forget to take other resource constraints into account either. Real-time assessment doesn't help if it takes an analyst a couple days to validate each alert and kick off the investigation process.

Another point to consider is the increasing overlap between real-time configuration assessment and the host intrusion prevention system (HIPS) capabilities built into endpoint protection suites. The HIPS is typically configured to catch configuration changes and usually brings along a more response-oriented process. That's why we called configuration management a periodic control in the Endpoint Security Management Buyer's Guide. That said, there is a clear role for configuration management technology in dealing with attacks and threats. It's a question of which technology – active HIPS, passive configuration management, or both – will work best in your environment.

In a perfect world you would like to continuously assess your devices, to shorten the window between attack-related configuration change and detection.

Managing Alerts

Given that many alerts from your configuration management system may indicate attacks, a key component of your operational process is handling these alerts and investigating each potential incident. We have done a lot of work documenting incident response fundamentals and more sophisticated network forensics, so check that research out for more detail. For configuration management a typical alert management process looks like this:

1. **Route alert:** The interface of your endpoint security management platform acts as the initial view into the potential issue. Part of the policy definition and implementation process is to set alerts based on conditions that you would want to investigate. Once the alert fires, someone needs to process it. Depending on the size of your organization that might be a help desk technician, someone on the endpoint operations team, or a security team member.

2. **Initial investigation:** The main responsibility of the tier 1 responder is to validate the issue. Was it a false positive, perhaps detecting an authorized change? If not, was it an innocent mistake that can be remedied with a quick fix or workaround? If not, and this is a real attack, then some kind of escalation is in order using your established incident handling process.

3. **Escalation:** At this point the next person in the chain will want as much information as possible about the situation. The configuration management system should be able to provide information on the device, the change(s) made, the user's history, and anything else that relates to the device. The more detail you can provide, the easier it will be to reconstruct what actually happened. If the responder works for the security team he or she can also dig into other data sources as needed — such as SIEM and firewall logs. At this point a broader initiative with specialized tools kicks in, and it is more than just a configuration management issue.

4. **Close:** Once the item is closed, you will likely want to generate a number of reports documenting what happened and the eventual resolution – at least to satisfy compliance requirements. But that shouldn't be the end of your closing step. We recommend a more detailed post-mortem meeting to thoroughly understand what happened, what needs to change to avoid similar situations in the future, and how processes stood up under fire. Critically assess the situation in terms of configuration management policies and make any necessary policy changes, as we will discuss later.

Troubleshooting

In terms of troubleshooting, as with patch management, the biggest configuration change risk is failure of a change. The troubleshooting process is similar to the one laid out in Patch Management Operations so we won't go through the whole thing. The key is to identify what failed, which typically involves either a server or agent failure. Don't forget about connectivity issues which can impact your ability to make configuration changes as well. Once the issue is addressed and the proper configuration changes made, you will want to confirm them.

Keep the need for aggressive discovery of new devices in mind — the longer a misconfigured device exists on your network the more likely it is to be exploited. As we discussed in the Endpoint Security Management Buyer's Guide, whether it's via periodic active scanning, passive scanning, integration with the CMDB (or another asset repository) or another method, you can't manage what you don't know exists. So focus on a timely and accurate ongoing discovery process.

Optimization

When you aren't dealing with an alert or a failure, you will periodically revisit policies and system operations with an eye toward optimizing them. That requires some introspection to critically assess what's working and what isn't. How long is it taking to identify configuration changes, and how is resolution time trending? If things move in the wrong direction, try to isolate the circumstances of the failure. Are the problems related to one of these areas?

- Devices or software
- Network connectivity or lack thereof
- Business units or specific employees

When reviewing policies, trends are your friend. When the system is working fine you can focus on trying to improve operations. Can you move, add, or change components to cut the time required for discovery and assessment? Look for incremental improvements and be sure to plan changes carefully. If you change too much at once it will be difficult to figure out what worked and what didn't.

Pay attention to maintenance of your environment. As with patch management, you need to keep the components updated and ensure that all updates are vetted before deployment.

Leveraging the Endpoint Security Management Platform

This paper has highlighted the intertwined nature of patch and configuration management. So we will wrap up by talking about leverage from using a common technology base (platform) for patching and configuration. Capabilities that can be used across both functions include:

- **Discovery:** You can't protect an endpoint (or any device, for that matter) if you don't know it exists. Once you get past the dashboard, the first key platform feature is discovery, which is leveraged across both patch and configuration management. The enemy of every security professional is surprise, so make sure you know about new devices as quickly as possible – including mobile devices.
- **Asset Repository:** Closely related to discovery is integration with an enterprise asset management system/CMDB to get a heads-up whenever a new device is provisioned. This is essential for monitoring and enforcement. You can learn about new devices proactively via integration or reactively via discovery — but either way you need to know what's out there.
- **Dashboard:** As the primary exposure to the technology, this is your view into the key operational processes being managed by the platform (like patch and/or configuration management). Using a single platform for both patch and configuration management, you will want the ability to only show certain elements, policies, and/or alerts to authorized users or groups, depending on their specific job functions. You will also want a broader cross-function view to track what's happening on an ongoing basis. With the current state of widget-based interface design, you can expect a highly customizable environment which lets each user configure what they need and how they want to see it.
- **Alert Management:** A security team is only as good as its last incident response, so alert management is critical. This allows administrators to monitor and manage policy violations which could represent a breach or failure to implement a patch.
- **System Administration:** You can expect the standard system status and administration capabilities within the platform, including user and group administration. Keep in mind that larger more distributed environments should have some kind of role-based access control (RBAC) and hierarchical management to manage access and entitlements for a variety of administrators with varied responsibilities.

- **Reporting:** As we mentioned in our discussion of specific controls, compliance tends to fund and drive these investments, so it is necessary to document their efficacy. That applies to both patch and configuration management, and both functions should be included in reports. Look for a mixture of customizable pre-built reports and tools to facilitate *ad hoc* reporting – both at the specific control level and across the entire platform.

Deployment Priorities

Assuming you decide to use the same platform for patch and configuration management, which capability should you deploy first? Or will you go with a big bang implementation: both simultaneously? That last question was a setup. As we've mentioned numerous times in the paper, we advocate a Quick Wins approach: deploy one function first and then move on to the next. Which should go first? That depends on your buying catalyst. Here are a few catalysts which drive the implementation of patch or configuration management:

We advocate a Quick Wins approach: deploy one function first and then move on to the next. Which should go first? That depends on your buying catalyst.

1. **Breach:** If you just had a breach, you will be under tremendous pressure to fix everything now, and spend whatever is required to get it done. As fun as it can be to get a ton of shiny gear drop-shipped and throw it all out there, it's the wrong thing to do. Patch and configuration management are operational processes, and without the right underlying processes the technology deployment will fail. If you traced the breach back to a failure to patch, by all means implement patch management first. Similarly, if a configuration error resulted in the loss, then start with configuration.

2. **Audit Deficiency:** The same concepts apply if the catalyst was a findings document from your auditor mandating patch and/or configuration. The good news is that you have time between assessments to get projects done, so you can be much more judicious in your rollout planning. As long as everything is done (or you have a good reason it isn't) by your next assessment you should be okay. All other things being equal, we tend to favor doing configuration management first, because configuration monitoring can alert you to compromised devices.

3. **Operational Efficiency:** If the deployment is to make your operations staff more efficient, you can't go wrong by deploying either patch or configuration first. Patch management tends to be more automated, so that's likely a path of least resistance to quick value. But either choice will provide significant operational efficiencies.

Summary

We have gone pretty deep into implementing and managing patch and configuration management — far deeper than most organizations ever need to get the technology up and running. We hope our comprehensive approach provides all the background you need to hit the ground running. Take what you need, skip the rest, and let us know how it works.

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/>.