Eliminate Surprises with Security Assurance and Testing

Version 1.6

Released: January 15, 2014

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.

This report is licensed by Ixia, whose support allows us to release it for free. All content was developed independently.

www.ixiacom.com

Ixia develops amazing products so its customers can connect the world. Ixia helps its customers provide an always-on user experience through fast, secure delivery of dynamic, connected technologies and services. Through actionable insights that accelerate and secure application and service delivery, Ixia's customers benefit from faster time to market, optimized application performance and higher-quality deployments. Learn more at http://www.ixiacom.com/solutions/security-test-tools/.

Copyright

Eliminating Surprises with Security Assurance and Testing

Table of Contents

No Surprises

The methods we use to develop and deploy applications and supporting infrastructure are undergoing fundamental change. Without diving into predictable hyperbole, new methods including DevOps and cloud computing promise to disrupt most of IT over the next 5-10 years. But embedded infrastructure and legacy applications are not going away. IT professionals need to walk a fine line between delivering critical services at the lowest price for acceptable performance, and doing it quickly and reliably.

As usual, security is at the end of the tail being wagged. It's hard enough to get developers to run a security scan on code before it is deployed into production. The idea of integrating security into these integrated development and operational processes (which Rich Mogull calls 'SecOps') seems like a pipe dream. It may be a dream today but it needs to become reality sooner rather than later.

IT has little choice. Adversaries continue to innovate and improve their tactics at an alarming rate. They have clear missions, typically involving exfiltrating critical information or impacting the availability of your technology resources. They have the patience and resources to achieve their missions by any means necessary. And it's your job to make sure deployment of new IT resources doesn't introduce unnecessary risk.

With this need to move faster and leverage more agile infrastructure, it is increasingly difficult to ensure adequate testing for infrastructure and applications before they go live. You have heard all the excuses that emerge when something goes wrong with a deployment.

- We didn't know it would get hit with that much traffic.
- We didn't get around to testing those edge cases.
- The application wasn't designed to do that, so that wasn't in the test plan.

The idea of integrating security into these integrated development and operational processes seems like a pipe dream. It may be a dream today but it needs to become reality sooner rather than later.

Ho hum, just another day in the office, and now it is Security's problem when the new application is compromised, data is lost, or the application falls over under the onslaught of a denial of service attack. It doesn't need to be this way. Really — it doesn't. Unfortunately common experience in the field has convinced many practitioners that security will inevitably take the blame.

The root cause of these issues is surprise. That's right — when an application goes live (or a major change goes into production), you don't really know what is about to happen, do you? You haven't been through a rigorous process to ensure the application and its infrastructure are ready for prime time.

Taking a page out of Google's handbook and calling the application 'Beta' won't save you. If the application has access to critical (regulated) information and is accessible — whether internally or externally — a security mindset is required, along with a way to put the application through its paces. As the Pragmatic CSO[1] explains,

> *Basically you are trying to eliminate surprises. So by doing a full battery of tests before the new system is deployed, you reduce the likelihood that you are missing something that you'll learn about later — the hard way.*

Technological disruption is not about to stop. If anything it will accelerate, so we need to get past our idea of a discrete security function *maybe* doing some testing and/or risk assessment at the tail end of a project. So what can and should security folks do? And how can they get both the development and operations teams on board with the necessary changes to ensure the protection and survivability of the application?

> Technological disruption is not about to stop. If anything it will accelerate, so we need to get past our idea of a discrete security function maybe doing some testing and/or risk assessment at the tail end of a project.

To avoid surprise we suggest a security assurance and testing process to ensure the environment is ready to cope with real traffic and real attacks. This goes well beyond what development organizations typically do to 'test' their applications, or ops does to 'test' their stacks.

It also is different from a risk assessment or a manual penetration test. Those "point in time" assessments aren't necessarily comprehensive. The testers may find a bunch of issues but they will miss some. So remediation decisions are made with incomplete information about the true attack surface of infrastructure and applications.

[1] http://www.pragmaticcso.com

This paper will dig into this Security Assurance and Testing process, discussing which devices and infrastructure components to test and how to consistently and reliably ensure you are testing their key functions. We will also focus on assuring the readiness and resilience of applications, because that's often the path of least resistance for attacks.

Remember — adversaries don't need to hit an arbitrary deadline. They will take the time needed to find the chinks in your armor. Maybe it's within the application, perhaps it's in the computing stack, or it could be the underlying equipment that gets data from one place to another. You cannot eliminate all the defects and security holes in your environment. But you can get a broader perspective of your exposures and put a plan together to protect it.

Tactics and Programs

As we discussed above, it is increasingly hard to adequately test infrastructure and applications before they go into production. Your adversaries have the benefit of being able to target the weakest part of your environment — whatever that may be. So the key to a Security Awareness and Testing (SA&T) process is to ensure you are comprehensively assessing your entire stack. Does that make the process much more detailed and complex? Absolutely, but without a comprehensive test you cannot be sure what will happen when facing real attackers.

To discuss tactics we consider how you would test your network and then your applications. We will also discuss testing exfiltration because preventing critical data from leaving your environment is another way to disrupt the Data Breach Triangle[2], which ultimately is the objective of the security team.

Testing Network Security

From security trade publications you would get the impression that attackers only target applications nowadays, and don't go after weaknesses in network or security equipment. *Au contraire* — attackers find the path of least resistance, whatever it is. So if you have a weak firewall ruleset or an easy-to-evade WAF and/or IPS, that's where they go. Advanced attackers are only as advanced as they need to be. If they can access your network via your firewall, evade your IPS, and then move laterally by jumping across VLANs… they will. And they can leave those 0-days on the shelf until they really need them.

If you have a weak firewall ruleset or an easy-to-evade WAF and/or IPS, that's where they go. Advanced attackers are only as advanced as they need to be.

So you need to test any device that sees the flow of data. That includes network switches, firewalls, IDS/IPS, web application firewalls, network-based malware detection gear, web filters, email security gateways, SSL VPN devices, etc. If it sees traffic it can be attacked, so you need to be ready.

So what should you actually test for network and security devices?

[2] https://securosis.com/blog/the-data-breach-triangle/

- **Scalability:** Spec sheets may be, um, inaccurate. Even if there is a shred of truth in the spec sheet, it might not be relevant in your configuration or with your application traffic. Make sure the devices will stand up to your *real* traffic at peak traffic volumes. Additionally, with increasingly common denial of service attacks, ensuring your infrastructure can withstand a volumetric attack is integral to maintaining availability of your infrastructure.

- **Evasion:** Similarly, if a network security device can be evaded, it doesn't matter how scalable or effective it is at blocking attacks. So ensure you are testing for standard evasion tactics as well.

- **Reconfiguration:** Finally, if the device can be reconfigured and unauthorized policy changes applied, the highly tuned and secure policies on your devices won't accomplish much. So make sure your devices cannot be accessed except by an authorized party using acceptable policy management.

Application Layer

Once you are confident the network and security devices will hold up move on to testing the application layer. Here are the highlights:

- **Profile inbound application traffic:** You do this to develop a baseline of application traffic and understand your normal volumes, protocols, and destinations. Then you can build scenarios that represent *not normal* use of the application to test edge cases. Be sure to capture actual application traffic so you can hide attacks within it, the way real attackers do. Obfuscation is a key tool in the attackers kit, so you need to think that way as well.

- **Application protection:** You will also want to stress test your WAF and other application protections using standard application attack techniques — including buffer overflows, application fuzzing, cross-site scripting, slow HTTP, and other application-centric denial of service[3] tactics. Again, the idea is to identify the breaking points of applications before your adversaries do.

The key to this assurance and testing process is to test as much of the application as you can. Similar to a Quality Assurance testing harness used by developers, you need to exercise as much of the code as you can to identify issues as early in the process as possible. Adversaries usually have time, so they will search every nook and cranny of your application to find its weak spot. If you aren't testing to the same level you are setting yourself up for failure.

Adversaries usually have time, so they will search every nook and cranny of your application to find its weak spot.

[3] https://securosis.com/research/publication/defending-against-application-denial-of-service-attacks

Exfiltration and Egress

The last aspect of your SA&T process is to see whether you can actually get data out. If the data cannot be exfiltrated it's not really a *breach*. Here you test content filtering capabilities — including DLP, web filters, email security, and any other controls that inspect content on the way out. Like the full code coverage approach discussed above, try to exfiltrate through as many applications and protocols as possible. That means all the major social networks (and some minor ones) and other logical applications like webmail. Also ensure you test encrypted traffic because attackers increasingly use multiple layers of encryption to exfiltrate.

Finally, test the feasibility of establishing connections with command and control (C&C) networks. These 'callbacks' identify compromised devices, so you will want to make sure you can detect this traffic before data is exfiltrated. This can involve sending traffic to known C&C nodes, as well as simulating communications with bot controllers using traffic patterns and domain generating algorithms typically associated with the behavior of compromised devices.

Test the feasibility of connecting to C&C networks by using traffic patterns and domain generating algorithms typically associated with the behavior of compromised devices.

The SA&T Program

We explained in our Continuous Security Monitoring[4] paper why you need to understand how ongoing (and never-ending) changes in your environment impact security posture. Similarly you need to think about SA&T from an ongoing perspective. To really understand how effective your controls will be, you need an SA&T program.

Frequency

The first set of decisions for establishing your program concerns testing frequency. The underlying network/security equipment and computing infrastructure tends not to change that often, so you can likely get away with testing these components less frequently — perhaps quarterly. But if your environment has constant infrastructure changes, or if you don't control your infrastructure (outsourced data center, etc.) you may want to test more often.

Another aspect of testing frequency is planning for *ad hoc* tests. This entails defining a set of catalysts that trigger a test. It could be something like a firewall rule change, which if done incorrectly could crush the performance of the firewall or open a hole in your perimeter big enough to drive a truck through. Or it might be a configuration change or device replacement that should trigger a broader test of the environment. Either way your SA&T program needs to specifically define when tests will occur, both for ongoing and *ad hoc* testing.

[4] https://securosis.com/research/publication/continuous-security-monitoring

Keeping Current

Nothing is static in today's environment. Attackers keep innovating and improving their attacks — so you need to keep your assurance and testing techniques, tactics, and technologies current. Practically, that means you need to ensure your testing systems and tools use the latest attack patterns and malware samples. That is the only way to know whether your controls will detect and block attacks. You can follow the latest and greatest attacks and malware yourself, or look to a threat intelligence service to keep you up to date. Either way staying current is critical to the success of your program.

Nothing is static in today's environment. Attackers keep innovating and improving their attacks — so you need to keep your assurance and testing techniques, tactics, and technologies current.

When a new attack is identified that could potentially exploit current defenses, that is a reasonable trigger for an *ad hoc* test as described above. We know of organizations that use a tabletop exercise when new attacks surface to get a gut check for whether the attack would have succeeded in their environments. But there is no substitute for the real deal, so when in doubt run a test to determine your true exposure.

Using Live Ammo

That brings us to a sensitive part of the SA&T program discussion: whether to run these tests in an isolated environment or against production systems. As with most things, the answer is a bit of both. Clearly running malware against production systems isn't a great idea. Okay, it's a bad idea. Testing defenses against specific attacks should take place an isolated environment (meaning a test environment), where you can control the spread of the malware and the attack can't actually access or accidentally exfiltrate sensitive data. However load impacts the behavior and availability of the content inspection devices, so make sure you test the environment with enough traffic simulating valid user behavior in the face of the attack volumes.

Should you run these tests in an isolated environment or against production systems? As with most things, the answer is a bit of both.

For testing evasion and availability attacks you won't build out a testbed at the same scale as your production environment, so you cannot effectively approximate how you will handle the attack unless you really attack the production environment. But be smart. Look for an underutilized window — perhaps in the middle of the night. Clearly that is inconvenient for you, sure. But finding that low activity window is much better for job security than testing during peak usage.

Specifically for volumetric Denial of Service[5] attacks, your program should include testing an actual transition to a scrubbing center, and back. You cannot really model or simulate this transition — you need to actually shift traffic and ensure systems remain available.

Bring on the Simians

You cannot model or simulate every permutation and combination of things that might happen. That's why we favor a comprehensive automated approach that covers as much attack surface as possible. We are huge fans of Netflix's Simian Army[6]. They use a set of tools to cause failures — forcing their systems, processes, and people to respond; and developing a much more resilient environment through constant practice and refinement of the recovery processes.

All these testing activities can be done manually, but you need automation to ensure consistency and accuracy over the long haul.

A similar approach can be used for SA&T. You might hide evasion attempts within a stream of legitimate traffic. You could blast your network with bursts of high-volume traffic at random times to simulate a surprise DoS attack. You can introduce vulnerable devices into the environment to see how long it takes to find them.

All these testing activities can be done manually, but you need automation to ensure consistency and accuracy over the long haul. Automating these testing actions is what makes Netflix's approach so interesting, and we believe in their foundational combination of automation and self-induced chaos (one of their 'simians' is actually called Chaos Monkey[7]).

The key is that you need a structured approach to testing your entire infrastructure on an ongoing basis. Many organizations just run simple scans against applications or devices to figure out whether a vulnerability exists. That's fine, especially in regulated environments with mandatory scanning, but scans won't necessarily show you what's really important. The important information to glean from an SA&T program is the weak spots in your business systems, and how to address those issues before attackers exploit them.

[5] https://securosis.com/research/publication/defending-against-denial-of-service-dos-attacks

[6] http://techblog.netflix.com/2011/07/netflix-simian-army.html

[7] http://techblog.netflix.com/2012/07/chaos-monkey-released-into-wild.html

Quick Wins

It is always helpful to see how the concepts we describe apply to more tangible scenarios, so we will show how an SA&T program can provide a *quick win* for the security team, with two (admittedly contrived) scenarios that show how SA&T can be used — both at the front end of a project and on an ongoing basis.

Infrastructure Upgrade

For our first scenario let's consider an organization moving to a private cloud environment to support a critical application, which is common these days. The business drivers are better utilization of data center resources and greater agility for deploying hardware resources to meet organizational needs.

Obviously this is a major departure from the historical rack and provision approach. It is attractive because it enables better operational orchestration, allowing for new devices ('instances' in cloud terminology) to be spun up and taken down automatically according to the application's scalability requirements. The private cloud architecture folks aren't totally deaf to security, so some virtualized security tools are implemented to enforce network segmentation within the data center and block attacks from insiders.

Without an SA&T program you would probably sign off on the architecture (which does provide some security) and move on to the next thing on your list. There would be no way to figure out whether the environment was *actually* secure until it went live, at which point attackers would let you know quickly enough. Using SA&T you can potentially identify issues at the beginning of implementation, saving everyone a bunch of heartburn. Let's enumerate some of the tests to get a feel for what you might find:

- **Infrastructure scalability:** You can capture network traffic to the application and use that traffic to test scalability. Keep in mind that simple traffic replay approaches tend to use a limited traffic sample over and over again, which doesn't really approximate real application traffic dynamics. So ensure the approach you use to test scale looks like *real* application sessions. After increasing traffic into the application you might find the cloud's auto-scaling capability inadequate. Or it might scale a bit *too* well — spinning up new instances too quickly or failing to take them down quickly enough. All these issues affect the value of the private cloud, and handling scaling properly can save a lot of heartburn for Ops.

- **Security scalability:** Another infrastructure aspect you can test is the scale of the security controls in use — especially for virtualized security tools. By blasting the environment with a ton of traffic you might discover that your virtual security tools crumble rather than scaling — perhaps because VMs cannot take advantage of custom silicon — and fall over. This failure normally either "fails open," allowing attacks, or "fails closed," impacting availability. You may need to change your network architecture to expose your security tools only to the amount of traffic they can handle. Either way, better to identify a potential bottleneck *before* it impairs either availability or security. A quick win for sure.
- **Security evasion:** You can also test security tools to see how they deal with evasion. If the new tools don't use the same policy as the perimeter, which has been tuned to effectively deal with evasion, the new virtual device may require substantial tuning to ensure security within the private cloud.
- **Network hopping:** Another feature of private clouds is their ability to define network traffic flows and segmentation — "Software Defined Networks." But if the virtual network is configured incorrectly it is possible to jump across logical segments to access protected information.
- **Vulnerability testing of new instances:** One of the really cool (and disruptive) aspects of cloud computing is elimination of the need for tuning configurations and patching. Just spin up a new instance — fully patched and configured — move the workload over, and take down the old one. But if new instances spin up with vulnerabilities or poor configurations, auto-scaling perpetuates security issues a lot faster than traditional infrastructure. Test new instances on an ongoing basis to ensure proper security. Again, a big win if you find something amiss before it bites you.

Many things can go wrong with any kind of infrastructure upgrade. A strong process to find breaking points in the infrastructure *before* going live can mitigate much of the deployment risk — especially if you are dealing with new equipment. Given the dynamic nature of technology, you will want to make sure you are testing the environment on an ongoing basis, as well as ensuring that change doesn't add unnecessary attack surface.

Many things can go wrong with any kind of infrastructure upgrade. A strong process to find breaking points in the infrastructure before going live can mitigate much of the deployment risk.

This scenario points out where many issues can be found. What happens if you can't find any? Does that impact the value of the SA&T program? Possibly, but don't discount the peace of mind from knowing the infrastructure is ready for production.

New Application Capabilities

To dig into another scenario, let's move up the stack a bit to discuss how SA&T applies to adding new capabilities within an application serving a large user community, to enable commerce on a web site. Business folks like to sell stuff so they like this kind of new capability. This initiative involves providing access to a critical data store from an Internet-facing application, which always raises concerns about data security.

The development team has run some scans against the application to identify application layer issues such as XSS, and fixed them before deployment by front-ending the application with a WAF. So a lot of the low-hanging fruit for application testing is gone. But that shouldn't be the end of testing. Let's look into some other areas where issues could crop up, focusing on realistic attack patterns and tactics:

- **Attack the stack:** You could use a slow HTTP attack to see whether the application can successfully defend against availability attacks on its stack. These attacks are very hard to detect at the network layer so you need to make sure the underlying stack is configured to deal with them.
- **Shopping cart attack:** Another type of availability attack uses the application's legitimate functionality against it. It's a bit like an autoimmune disease, where the application is attacked by its own capabilities. By overwhelming the shopping cart (with automation), attackers overwhelm the site as it tries to provide billions of search results — again impacting response time and application availability, which is bad for a commerce application (or almost any application, really).
- **WAF evasion:** You can go significantly deeper into an application by trying to evade defenses in front of it, such as a WAF. We mentioned above that an application security scanner showed no issues with XSS because the WAF blocked it. By easily evading the WAF, a more sophisticated testing capability showed the application was a sitting duck for buffer overflow and XSS attacks. A simple app scanning tool cannot provide that kind of result unless run on the internal network, which doesn't reflect the true attack surface as viewed by the adversary.
- **Go after the data:** By compromising different parts of the application stack — such as application servers — attackers can position themselves to attack the data store. As part of the testing process, try to find and address the data store directly. Many database security controls run in front of the database to avoid impacting performance of the DBMS, so if an attacker can figure out how to connect directly to the database it might be "Game Over."

As with fully testing the infrastructure, a deeper level of application testing may reveal issues that could cause downtime or data loss. The challenging part of application security testing is that attackers can use legitimate functionality — including search and shopping carts — to attack, making it even harder to determine true production readiness.

But you aren't done yet. As mentioned above, you can and should also be running frequent exfiltration attacks to determine whether data can be removed from your network. This ongoing testing assesses the vigilance of your security team and the effectiveness of your controls and processes. Alerts streaming from content filtering tools are all well and good, but unless someone is validating each alert and determining the extent of any potential data loss, the alert is worthless. So an SA&T program not only tests technology underpinnings, but the real effectiveness of the security function.

The challenging part of application security testing is that attackers can use legitimate functionality — including search and shopping carts — to attack, making it even harder to determine true production readiness.

Handling the Truth

These kinds of findings from an SA&T program may be controversial within the organization — especially if Operations or Development is under significant pressure to complete the project. Testing by another group may be perceived as a roadblock, getting in the way of delivering value to customers and generating revenue. That is all too common — some leaders may resist holding up the project to address issues you find. The only way to address this resistance is with information. By being very clear about the downside (risk) of identified issues, you can enable business leaders to make business decisions about whether to accept risk and move forward, or hold up until the issues are addressed.

Keep the SA&T program in perspective. If leadership decides to move forward anyway — despite significant issues — that is not a program failure. The goal of SA&T is to provide *information* to make *educated* decisions. A program should not be evaluated based on whether its findings are ignored, or the ultimate consequences of those decisions. Just make sure to document the findings so the team can rise above finger-pointing if issues do crop up.

The goal of SA&T is to provide information to make educated decisions. A program should not be evaluated based on whether its findings are ignored, or the consequences of those decisions.

Summary

As we wrap up Security Assurance and Testing, let's highlight a few key points. First, you don't know what is truly at risk until you actually attack yourself. Attackers don't play nice, so tabletop exercises, threat models, and simple vulnerability scanners are inadequate to assess your infrastructure for the information you need to *really* understand how your environment can be exploited.

Don't forget to pay attention to the network/security devices your technology infrastructure is built on. Attackers take the path of least resistance, so if your infrastructure is brittle they will figure that out and take the easy route. Then they get to save their fancy advanced 0-day attacks for an environment that deserves it.

Finally, the SA&T program involves ongoing testing over as much of the environment as possible, including infrastructure and applications. Make sure you are testing for both availability/scalability and evasion of controls, because both provide opportunity for an attacker to gain presence in your environment. The SA&T program should include testing both on an ongoing basis and *ad hoc*, when the environment undergoes significant change.

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 <https://nexus.securosis.com/>.

About the Analyst

Mike Rothman, Analyst/President

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

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

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

About Securosis

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

- **The Securosis Nexus**: The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics 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: <https://securosis.com/>.