Rethinking Security in the Era of Cloud Computing

Jay Aikat | University of North Carolina at Chapel Hill
Aditya Akella | University of Wisconsin at Madison
Jeffrey S. Chase | Duke University
Ari Juels | Cornell Tech
Michael K. Reiter | University of North Carolina at Chapel Hill
Thomas Ristenpart | Cornell Tech
Vyas Sekar | Carnegie Mellon University
Michael Swift | University of Wisconsin at Madison

Cloud computing has emerged as a dominant computing platform for the foreseeable future, disrupting the way we build and deploy software. This disruption offers a rare opportunity to integrate new computer security approaches.

Much has been made about the dramatic growth of compute clouds and their transformative role in modern computing. Cloud computing's scale suggests that it will be a driving force for the foreseeable future; a significant and growing fraction of the web is now hosted in major cloud operators' datacenters.[1] The often cited reasons for their success include the cost savings and flexibility they enable by freeing tenants from the expenditures of establishing and operating their own datacenters.

A complicating factor in the rush to cloud computing is tenant security. For several reasons, security risks can be exacerbated by moving to the cloud. As cloud computing grows in popularity, clouds become juicier targets for attackers. Because the cloud puts tenants on a common computing infrastructure, one flaw exposed in that infrastructure can threaten all tenants. Clouds gather computations of potentially mutually distrusting tenants on the same compute platform, potentially introducing opportunities for cross-tenant attacks. These risks have given rise to numerous active and worthwhile research agendas to enable tenants to mitigate them.

We believe there's another research agenda that the move to cloud computing should ignite but that so far has received less attention from the research community. The consolidation of massive resources and myriad activities in the cloud places cloud operators in a unique position to introduce new services to help tenants better manage their security (or to manage it for them outright)—and indeed to solve some of security's "holy grail" problems. Starting in 2013, we were funded by the NSF to explore these opportunities; our goal in this article is to summarize our research agenda and the progress we've made so far. We refer to our effort as the Silver project, connoting a silver lining to the cloud.

Cloud Security Research Themes

Here we provide a brief overview of the cloud security research landscape, roughly drawing a dichotomy between two research motivations. The first, which has been the primary concern of academics thus far, is what new threats to tenants arise due to their move to cloud computing, and what tenants can do to mitigate those threats. We then juxtapose this with another area of research that we find understudied—that of enhancing clouds to help tenants achieve secure deployments. (For a brief history of clouds, see the sidebar.)

Risks from Cloud Computing, and How Tenants Can Mitigate Them

Much work has focused on determining how the shift to cloud computing and public clouds might create new vulnerabilities for tenants. In one type of new risk, a cross-tenant attack arises when one cloud tenant seeks to violate the confidentiality, availability, or integrity of another. Many cross-tenant attacks stem from the fact that multitenancy implies that it's possible that one's

1540-7993/17/$33.00 © 2017 IEEE

A Brief History of Clouds

Modern cloud computing systems are the latest example of shared-infrastructure computing. The idea of "utility computing" on mainframes was a primary motivation for the Multics project, which inspired much early research on computer security. In such systems, users and businesses remotely accessed a central mainframe, amortizing the large cost over many users. With the rise of the World Wide Web in the late 1990s, public hosting centers grew in popularity and offered both bare-metal machines and managed webservers. The free Linux OS and Apache webserver, launched in 1991 and 1995, respectively, and servers with commodity Intel and AMD processors, made it economical to run large server clusters to handle the increasing scale of Internet workloads.

Two innovations really drove the cloud revolution. First, VMware introduced a hypervisor for x86 processors, showing that virtualization could be efficient and useful. This was followed by the open source Xen project from Cambridge University in 2003. Second, Amazon launched the Elastic Compute Cloud that rented virtual machines with a new billing model based on short-term usage. For a few cents, a customer could rent a fraction of a physical machine for an hour. The combination of virtualization, efficient management, and short time-scale billing enabled cloud computing's boom.

Today's public clouds provide access to large-scale computing infrastructure in the form of virtual machines, applications, or software services. The common thread among all these is the dynamic sharing of infrastructure by multiple tenants, which is managed by a third party. In this common structure, clouds today can be divided roughly into three kinds. Infrastructure-as-a-service systems allow customers to launch virtual machines and control the guest OS as well as applications. In platform-as-a-service systems, customers can launch applications that execute in a provider-managed OS. Finally, a software-as-a-service cloud provides hosted application services for customers, such as email, databases, file storage, and the like. A single company, such as Google or Microsoft, might offer all three kinds of computing. The division into these categories follows the level within a typical software platform at which there is a management handoff from the cloud provider to the customer.

applications are running on a compute server shared with another tenant who might be hostile. Placement vulnerabilities allow malicious tenants to arrange for their applications to be scheduled (placed) on the same server as a target,[2–4] and from there, cross-tenant side-channel attacks could allow violating isolation boundaries to, for example, steal secrets[5,6] or computing resources,[7] or to mount denial-of-service (DoS) attacks.

A second type of new risk is from a cloud operator who is itself malicious. This might seem, at first glance, to be a strange perspective: Why would one use a cloud that's adversarial? In fact, this threat model covers many nuanced and realistic threats, such as a rogue employee's insider attack and even a careless provider's accidental errors in the handling of customer data and programs. The widespread occurrence of malicious outsiders intruding into the systems of even large, highly reputed technology companies also encourages such a pessimistic threat model.

These new risks have given rise to several influential research threads, notably works on cryptographic techniques such as verifiable outsourced computing and outsourced private computation via fully homomorphic encryption.[8,9] These lines of work strive to remove the cloud provider's software and administrators from the trusted computing base. However, this threat model inherently pushes more responsibility (at the very least, for cryptographic key management) from the security-conscious providers to tenants that are, on the whole, entirely unsophisticated when it comes to security. Another drawback to such approaches is their high performance overhead, which will remain even if innovations bring about orders-of-magnitude improvements.

Another Viewpoint: Provider as Partner

The above research agendas can be viewed as helping tenants to both understand and preempt new vulnerabilities that arise when moving to a cloud that's oblivious to its tenants' plights at best and openly hostile to them at worst. We believe these agendas are important to pursue further, but they also overlook a largely untapped opportunity: Can security-literate cloud providers help security-illiterate tenants resist threats? This viewpoint is already widely adopted in industry, where companies increasingly offer security-as-a-service solutions. In general, however, these solutions simply represent adoption of traditional security mechanisms in noncloud settings. Although default availability of "best practices" in security would mark important progress and make deployment and management simpler, we fear that the cloud computing industry isn't currently taking full advantage of the security opportunities that the shift to cloud computing offers.

These opportunities stem from the strategic position occupied by service providers to manage security on their tenants' behalf. In particular, providers can take advantage of deeper specialization and introspection, a broader perspective, and more compute resources.

Deeper specialization. A recurring obstacle to operational security in practice is a dearth of professionals with relevant expertise. Cloud-based security services allow many organizations to benefit from the deeper specialization of a few large cloud providers and security service providers.

Provider introspection. Cloud providers have the unique ability to analyze tenants' security, which we argue helps facilitate outsourced security management. One (well-known) challenge to taking advantage of introspection is the semantic gap between the provider's observations and the significance of observed behaviors in tenant environments, which might be highly customized. Another challenge is cloud providers' reluctance to access tenant data, due to confidentiality concerns stemming from the sensitivity of tenant data and from regulatory requirements around data sharing. But even lightweight introspection could yield big rewards in terms of improved security.

A broad view. Cloud services attract a large number of customers, enabling cloud providers to obtain a broad view of security-relevant information. Aggregating this information could lead to deeper insights than those obtainable by smaller organizations in isolation—a form of "herd immunity" to threats.

Massive compute resources. Security analysis of the information knowable to cloud operators might require massive compute and storage resources. But large cloud providers have access to these resources.

These strategic benefits are somewhat at odds with the other common cloud research agendas we mentioned earlier. For example, many of the research agendas that presume malicious providers seek to prevent all introspection on tenant applications. We believe that provider interaction with tenant environments is not merely a threat but also an opportunity to improve tenant security.

A Silver Lining to the Cloud

Here we lay out one vision of how clouds could significantly enhance security for their tenants. We dissect this vision into three distinct but overlapping opportunities for leveraging the cloud to help tenants manage their untrusted client populations, their own outsourced infrastructure, and their dependencies on other tenants in the cloud ecosystem.

Helping Tenants Manage Clients Securely

A substantial fraction of the effort to run an application service is consumed by addressing the risks posed by untrusted clients. These risks include client account takeovers, DoS attacks, and exploit attempts on the server. We see numerous opportunities to leverage the cloud's massive resources and elastic scaling capabilities to help tenant services defend against these risks. We outline several examples below.

Authenticating clients of tenant services. Authenticating human users is central to virtually every web transaction. Overwhelmingly, the most common way to do this today is using passwords. Passwords are an imperfect protection, however. They can be guessed by an attacker or, as today's industry pandemic shows, they can be stolen by breaching the application server and cracking password hashes stored there.

We describe two approaches by which a cloud operator can aid tenants in authenticating clients more securely: Pythia and honeywords. These solutions aim to minimize the risk of an undetected password database breach or user impersonation attempt. They can be used by ordinary tenants to protect their authentication systems but are especially attractive as a means to strengthen the identity service providers' infrastructure. They can also facilitate compliance with security controls for identity and access management.

Pythia allows a cloud operator to harden passwords on an application server, protecting them against compromise during a breach.[10] It's transparent to application service users, imposes minimal additional latency, and requires only minor modifications to the application server.

Pythia relies on a pseudorandom function (PRF) server, potentially run by the cloud operator. The server applies a PRF—a deterministic cryptographic operation involving a secret key—to a password p that the application server submits for registration (storage) or verification, yielding a corresponding output x. The PRF server thereby "hardens" passwords. Unlike a password hash, which is vulnerable to brute-force cracking if the underlying password is weak, the PRF-hardened value x is computationally infeasible for an adversary to crack.

Pythia provides additional security features. It's partially oblivious—passwords submitted to the PRF server are cryptographically concealed from the cloud operator, yet Pythia still enables the cloud operator to perform account-level monitoring of authentication attempts. Pythia also supports efficient key updates, meaning that the PRF server can update a PRF key and send a compact update token to the application server that permits every user's hardened password representation x to be updated accordingly. Such key rotation nullifies the effects of a breach of the application server (or the PRF server).

One limitation of Pythia is that, to detect a breach, it relies on anomaly detection—an error-prone approach.

In contrast, a scheme we've developed called honeywords aims to detect password system breaches.[11]

A honeyword is a fake but plausible-looking password. The honeywords scheme involves storing for each user in the password database on an application server, not just the user's real (hashed) password *p*, but also a set of $n - 1$ (hashed) honeywords. An adversary that breaches the server faces the challenge of distinguishing the unique true password from the honeywords. If the adversary guesses incorrectly and attempts to authenticate using a honeyword, an alarm is triggered. Given good honeyword selection, an adversary will evade detection with a probability of only $1/n$.

To give further detail, the full set of *n* passwords, real and fake, is stored on the application server in a randomly permuted list. When an authentication attempt occurs using a password in the list associated with a given user, the associated index is passed to a system maintained by the cloud operator called a honeychecker. The honeychecker stores the index (position) of the unique true password for every user and distinguishes between the true password and honeyword submissions. Thus, the honeyword system (like Pythia) achieves breach resistance by spanning the tenant and the cloud operator environments.

> **By focusing on the root cause of side channels (that is, coresidency), Nomad is agnostic to the specific side-channel vector used.**

Later, we discuss other intriguing uses of *honey objects*, that is, fake resources for breach detection and adversarial misdirection in the cloud.

Fending off DoS attacks against tenant services. A second threat that administrators of application servers must address is DoS attacks against their servers. The damage that DoS attacks cause to organizations in terms of lost revenue and customer trust is well known. DoS defense of application servers today comes primarily in two forms: proprietary solutions that scale to massive load but that are expensive (for instance, Akamai) or hardware appliances that will have difficulty keeping up with adversaries' ability to dynamically change their attacks' type, volume, and locations.

As part of the Silver project, we envision a cloud-based DoS defense architecture that provides the flexibility to seamlessly place defense mechanisms where they're needed and the elasticity to launch defenses as necessary depending on the attack type and scale. As a proof of concept, we developed the Bohatei system, which leverages several of the advanced software-defined networking and network function virtualization capabilities.[12] Specifically, Bohatei leverages these capabilities to elastically adapt the scale and type of defenses needed and to steer suspicious traffic through the defenses deployed at suitable cloud locations.

Detecting exploit traffic against tenant services. Another threat type that administrators of application servers need to constantly fend off is exploit attempts against their servers. Sometimes these exploit attempts target logic vulnerabilities in the application servers themselves; in others, they target component protocols that the application servers employ. In many cases, exploits against such vulnerabilities involve clients sending traffic that no legitimate client implementation would send. Examples of such exploits include 10 Common Vulnerabilities and Exposures (CVEs) since 2014 for OpenSSL alone, including the well-known Heartbleed vulnerability (CVE-2014-0160).

We're developing a technique that leverages the cloud's spare compute resources and its visibility across tenant services to detect the emergence of new exploits as soon as they're attempted against any tenant. A cloud resident verifier analyzes the messages between a tenant server and its clients to detect messaging behavior from the client that's inconsistent with the expected client software. For example, our verifier can detect a client's deviation from an OpenSSL implementation of TLS within seconds from when the deviation occurs.[13] Because such deviations are typically characteristic of maliciously crafted packets to exploit server vulnerabilities, this type of verification capability could reduce the delay to detect exploit attempts on zero-day vulnerabilities. For example, this technique could have detected Heartbleed packets within seconds of the first attempted exploit, with no Heartbleed-specific configuration.

Helping Tenants Manage Their Infrastructure

Numerous organizations outsource portions of their own IT infrastructure to clouds, even if only to serve intraorganization purposes while achieving the cost savings associated with cloud computing. We present several ways in which cloud operators could assist tenants in managing the security of their outsourced infrastructure.

Side-channel defense. The basis for the dramatic cost savings enabled by clouds is the sharing of the resources that they enable so effectively. However, this sharing doesn't come without consequences. As we discussed, research has shown that sharing hardware resources can cause the unintentional leakage of secret information across tenant boundaries in cloud contexts. These leakages, called side channels, arise due to tenants' shared use of microarchitectural components on the computers they occupy together. Indeed, such side channels seem inevitable on current hardware platforms, if left unchecked by other defenses.

As a result, in Silver we're leading the development of operator-supported defenses against side channels in cloud contexts, ranging from specialized defenses against side channels in processor caches to more holistic defenses for wide ranges of side-channel attacks arising from coresidency. An example of a cache-specific defense is hypervisor scheduler modifications to ensure that one virtual machine (VM) can't preempt another with very fine granularity,[14] as this is an ingredient in known side-channel attacks leveraging per-core caches. Nomad is an example of a more holistic defense that extends this idea via a provider-assisted service that limits cross-tenant information leakage by carefully coordinating the placement and migration of VMs.[15] By focusing on the root cause of side channels (that is, coresidency), Nomad is agnostic to the specific side-channel vector used.

Tenant networking. Today, many organizations rely heavily on network functions (NFs) or middleboxes to implement sophisticated security functionality, including intrusion detection and prevention, monitoring, deep-packet inspection, and firewalls. Often, many of these functions are "chained," with traffic flows routed along a sequence of these middleboxes to enforce key security policies. As these organizations move their compute infrastructures into public clouds, it becomes important to realize equivalent functionality in these new settings. To facilitate this, our work is leveraging key properties of the cloud infrastructure, namely that it's virtualized and software defined. We argue that this not only enables flexible realization of the above functionality but also allows new functionality to be realized in the cloud context.

Specifically, we've developed three frameworks to illustrate these possibilities. The first, FlowTags, allows flexible routing of traffic across arbitrary chains of middleboxes, even as middleboxes alter packet header information on which routing traditionally relies.[16] FlowTags achieves this by inserting tags into end-to-end flows; the logic for computing tags resides at a logically central controller that leverages high-level policy to determine how the tags encode required end-to-end paths and any middlebox-internal actions taken along a route (for instance, content being served out of a cache). The tags can be consumed by middleboxes, enabling them to process the context of processing received by a flow so far (for instance, that it traversed a specific set of middleboxes). This allows end-to-end traffic-steering policies to be implemented for specific sets of flows.

The second system, OpenNF, is complementary to FlowTags and designed specifically to support distributed processing across multiple middlebox instances.[17] Although virtualization of middleboxes allows easy deployment and teardown of instances in a cloud setting, reallocation of processing across middlebox instances must be coordinated with reallocation of the internal state that middleboxes maintain for the traffic they're processing. OpenNF allows such state reallocation to be synchronized with traffic reallocation decisions and to take place safely and consistently. This capability enables novel security applications, for instance, a security application whose capability can be dynamically enhanced to detect sophisticated attacks. That is, an on-site middlebox (for example, a deep-packet inspection engine) employs simple, local processing to identify suspicious traffic access patterns. When such patterns are observed, deeper analysis of those patterns is seamlessly migrated (along with the state created so far by the local instance's processing) to a more capable, cloud-resident appliance.

The third system, Policy Graph Abstraction (PGA), offers the capability to effectively utilize the other two.[18] In many organizations, policies are independently specified by different actors; for instance, department administrators might want to restrict access to servers they own to users with specific credentials, whereas an enterprise-wide policy might impose general constraints on who can access what resources. It's important to ensure that such independently specified policies are composed and implemented consistently in the underlying infrastructure. PGA provides operators a simple graphical interface to specify complex policies among different sets of end points, including policies on middlebox traversal and elastic scaling. Each policy can be supported individually using FlowTags and OpenNF.

Crucially, the PGA runtime analyzes multiple such policies for potential conflicts and, if none exist, quickly computes a routing configuration that ensures consistent policy enforcement.

Strengthening Tenant Ecosystems

Another research direction is to explore how cloud providers can broker trust among tenants. Cloud platforms continue to advance their offerings of foundational services, for example, managed storage, coordination and consensus, and security-enhancing services. We envision that new cloud services can help to mediate secure interactions among tenants and further enable tenants to offer secure foundational services and application services to one another, even without a priori trust in the service owner. Flexible trust management, rooted in shared trust in cloud infrastructure providers, can enhance the potential for an open marketplace of cloud-based services.

One motivating scenario is to enable secure sharing of data and code for cooperative analytics, in which analytics software is offered as a service for computing with datasets or algorithms that their owners consider confidential. A goal is to enable jointly trusted computations that combine confidential datasets from multiple owners and produce a privacy-preserving output. Safe data sharing can help unlock the potential of big data in areas where data privacy is paramount, for instance, healthcare. We're developing cloud-based technologies to place such collaborations on more secure foundations.

> **Flexible trust management, rooted in shared trust in cloud infrastructure providers, can enhance the potential for an open marketplace of cloud-based services.**

A mutually trusted cloud provider can enhance security by mediating these kinds of cross-tenant interactions. One obvious way to approach these goals is to extend cloud authorization models to allow richer policy control over data sharing and other interactions among tenants. A more ambitious direction is to establish new cloud-based trust services that enable clients to derive trust in a tenant service without prior knowledge or trust in the identity of its owner. In our approach, the cloud provider serves as a root of trust by certifying facts about the tenant security properties or code identity. Other tenants might specify trust policies to evaluate against these assertions. For example, a data owner might trust a third-party service to access a sensitive dataset if the cloud provider attests that the service is contained—it's restricted in how it can communicate or release information.

Cloud support for contained execution. We're developing an extended infrastructure-as-a-service (IaaS)-layer framework for managing contained execution, in which a group of tenant instances (VMs) has its network connectivity restricted according to a declared policy as a defense against information leakage. One system prototype, called CQSTR (pronounced "sequester"), implements a new cloud container abstraction as a set of extensions to the OpenStack IaaS platform.[19]

A CQSTR cloud container is a grouping of VM instances comprising an application deployment. A cloud container specifies containment properties that limit network and storage access for computations in the container. CQSTR modifies existing IaaS-level management services to ensure that backups, log monitoring, and other management services can't be abused to extract data from a closed container. In addition, the policy could specify the set of images that's allowable to boot VM instances into the container: the cloud platform provides a simplified form of code attestation using the cloud provider as a root of trust, rather than a hardware root of trust (for example, a Trusted Platform Module). Attesting to a limited set of boot images enables a client to ensure that a service runs on a patched, locked-down OS and a trusted application framework that might implement additional security controls.

We've experimented with several application scenarios that use cloud containers for secure analytics. In these scenarios, CQSTR enables a data owner to enforce control over how its data is used by an analytics service. The owner can demand and verify that data is held securely in a cloud container that's safe from data leakage and misuse. With a cloud container, code interacting with a service can contact CQSTR to verify the service's containment and code properties in advance. A data owner can specify access-control lists (ACLs) with the containment properties needed to access the data. CQSTR extends IaaS storage services to be aware of cloud containers: storage services can base access control on the declared container properties governing the calling VM instance, according to the ACL policies specified by the data owner.

Building trust in tenant services. CQSTR is just one example of a cloud provider service that makes trusted statements about the security properties of a tenant's configuration and allows other client software to check these properties for compliance with a security policy. Other useful security properties available to the cloud platform include a tenant's firewall posture, whether its software is patched adequately, whether it runs defensive (for instance, antivirus) software, whether it encrypts its stored data, whether its password system is protected by Pythia, attestations of software identity, network security services, and so forth. They might also reflect continuous auditing or monitoring checks or incident history. Clients can use this information to make informed decisions about whether a service is trustworthy, rather than relying merely on its reputation, as is common today.

Our research seeks to provide a general and practical foundation to manage trust in cloud ecosystems based on authenticated assertions and policies, building on a wealth of prior work in authorization logics. Our approach addresses several potential concerns. First, the policy language should be expressive, efficient, easy to use, and extensible to a growing vocabulary of security properties. Second, the approach should protect the secrecy of sensitive data, including security configurations and the policies themselves.

We developed a declarative logic-based language (a trust logic) and interpreter software (called SAFE) to enable participating entities—including services of the cloud providers, tenants, external services, and client software for end users—to issue authenticated assertions about one another, and reason from the assertions of others. The language also expresses logical policy rules, which are verifiable. Declarative policy enables brokered compliance checks, in which a client submits a declarative security policy to a trusted policy engine (interpreter) that's operated—or attested—by the cloud platform: the policy engine checks compliance with the policy without revealing the policy or the security properties to anyone. A privacy-preserving compliance intermediary is an example of a secure foundational service for mediating tenant interactions in the cloud.

SAFE is suitable for more general trust management in federated environments, including systems spanning multiple networked cloud providers (for instance, ExoGENI). In this case, the participants can exchange SAFE security assertions and policy rules as signed certificates and run a local off-the-shelf interpreter to generate proofs of policy compliance end to end. SAFE also serves as a basis for more general access control in networked cloud systems. For example, we've implemented a reusable package of rules for nested groups and roles in a secure hierarchical name space, equivalent in power to the naming and authorization structure of Amazon Web Services (AWS) Identity and Access Management but applicable to multidomain systems rather than relying on a single trust anchor. SAFE can also support rich policies to authorize interconnection among tenants and connectivity with external networks.

Securing the PaaS layer. Increasingly, commercial cloud operators offer higher-level platform abstractions (platform-as-a-service, or PaaS) for tenants, for instance, Google's AppEngine and AWS Elastic MapReduce. PaaS systems simplify cloud programming with more powerful models that enhance customer productivity and add value to cloud services. PaaS systems are also offered by tenants to other tenants (for instance, Heroku and CloudFoundry).

Flexibility to offer layered platforms is fundamental to an open cloud ecosystem. We're exploring how to enable deployment of third-party PaaS services that inherit trust from the underlying IaaS cloud system via attestation, auditing, and mediated access to security credentials for IaaS-layer services. The PaaS service, in turn, could leverage its higher-level programming model to enforce language-based safety checks or interpose higher-level monitoring or containment. In addition, we believe that PaaS platforms are promising targets for trustworthy computing via software attestation and code-based access control. Our premise is that higher-level PaaS programs are more practical to verify and attest than binary executables because they're compact: they build on powerful languages and a library of standard primitives whose implementations can be trusted.

As one example, we're developing minimal trust extensions to a standard Spark analytics stack (spark.apache.org) to provide a PaaS service for secure cooperative analytics. It offers rich access control that allows data owners to regulate data sharing with other tenants on their own terms. The PaaS service attests to code identity for the stages of the analytics workflow: parties can designate mutually trusted Spark programs that can input sensitive data, possibly from multiple owners, and generate "declassified" outputs that are safe to share. The system tracks flow through the workflow to ensure that the security label for each object reflects its contents' potential sensitivity.

Industry-Inspired Challenges

We asked colleagues in industry to enumerate the cloud security problems they consider most pressing for their operational and business needs. Their responses not only present a range of open challenges and opportunities but also help illuminate ways that the viewpoint of "provider as trusted partner" can bring about further advances in cloud security. We offer a few examples here.

Securing Security Logs

Security logs are important for forensic investigation of security events and increasingly also in security analytics systems, which aim to detect anomalous events indicative of security compromises. Industry practitioners have stressed the importance of ensuring the integrity of security logs; the ability to extract meaningful intelligence from them remains a perennial technical challenge.

Although forward-secure cryptography has been advocated for securing logs in a device, it can only detect tampering, not prevent it. However, cloud operators could provide several services for securing logs against tampering despite compromise of a tenant's operational environment. It could, for example, provide a real-time pipe for logs, quickly removing logs from a tenant's environment for storage in an operator-secured environment. The operator could additionally enrich logs to improve their utility in security analytics. For example, the provider could offer a uniform timestamping service, which would address some of the fundamental synchronization issues identified in today's logging systems.

Furthermore, capitalizing on its broad view, a cloud operator could perform security analytics over the log data of multiple tenants, enabling identification of broad attacks and differentiation of targeted attacks through comparison. The major impediments to this opportunity are the challenges of scaling and, more fundamentally, the need to protect tenants' confidentiality: although a tenant can perform security analytics on its own data, security analytics encompassing multiple potentially mutually mistrusting organizations would require computation over combined data. Cloud support for contained execution offers a promising approach.

> **There's an opportunity to deploy honey objects in a tenant's environment with the cloud operator's support.**

Security Control and Vulnerability Mapping

Another opportunity in cloud systems that industry practitioners have identified is the need to map security controls, as defined in information security standards, such as ISO/IEC 27001:2013 and NIST Special Publication SP 800-53, to policies and technical enforcement mechanisms. In most organizations, this process is a time-consuming and labor-intensive business requirement; it involves inventorying systems and assets in conjunction with a systematic review of information security policies. However, a cloud operator—particularly one offering security services to its tenants—can in principle leverage economies of scale and homogeneous elements of its tenants' environments to streamline such mappings.

As an example, the Cloud Security Alliance Cloud Controls Matrix v. 3.0.1 control IAM-01, Identity & Access Management: Audit Tools Access, specifies that "Access to, and use of, audit tools that interact with the organization's information systems shall be appropriately segregated and access restricted to prevent inappropriate disclosure and tampering of log data." In implementing protections for security logs such as those we described earlier, a cloud operator can facilitate compliance with controls of this type by providing management tools for tenants and/or incorporating compliant controls into applications that the operator itself provides. A further opportunity exists when software vulnerabilities are identified. An organization must then identify affected systems and formulate and prioritize remediation plans.

Tools such as Amazon CloudTrail (for logging) and Amazon Inspector go some way toward realizing this vision of automated control/vulnerability mapping but are in their infancy and limited in scope. For example, CloudTrail handles only AWS API calls. Significant opportunities exist in extending the reach and sophistication of these tools.

Thwarting Adversarial Reconnaissance

A hallmark of recent advanced persistent threats is their apparent reliance on extensive preliminary reconnaissance. Industry practitioners have thus highlighted the need for cloud-based resources to mitigate reconnaissance, particularly given the risks that aggregation of organizations in the cloud presents.

One opportunity here lies in counterintelligence using decoy or honey objects. Honeywords is one example of such objects. But there's a much more general opportunity to deploy honey objects in a tenant's environment with the cloud operator's support. The cloud operator could create and monitor honey files or documents, honeypots, honeytokens, or honeynets.[20]

Cloud operator support of honey objects offers several attractive features. First, a cloud operator can maintain state outside a tenant's environment that distinguishes between real and honey objects; such state can thus be protected from compromises that occur inside the tenant's environment. Second, the cloud operator can observe adversarial interaction with honey objects via introspection, enabling well-concealed breach detection and monitoring of adversarial behavior. Finally, the cloud operator is well positioned to take an active role in honey object deployment. Thanks to the cloud's elasticity, servers and entire networks can quickly be spun up upon suspicion of compromise or in a way tailored to the apparent behavior of an adversary, enabling real-time deployment of strategies for misdirection or provision of misinformation to the adversary.

Cloud computing has unquestionably transformed the computing landscape. We believe, however, that opportunities for further transformation made possible by the move to clouds remain underexplored, particularly leveraging clouds to improve their tenants' security. We have contrasted this research vision to the dominant trends in security research motivated by cloud computing today. We also summarized several of the research directions we are exploring within the context of this vision as well as several additional opportunities identified through conversations with members of the cloud operator and tenant communities. ■

References

1. K. He et al., "Next Stop, the Cloud: Understanding Modern Web Service Deployment in EC2 and Azure," *Proc. Conf. Internet Measurement Conference* (IMC 13), 2013, pp. 177–190.
2. T. Ristenpart et al., "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," *Proc. 16th ACM Conf. Computer and Communications Security* (CCS 09), 2009, pp. 199–212.
3. V. Varadarajan et al., "A Placement Vulnerability Study in Multi-tenant Public Clouds," *Proc. 24th USENIX Security Symp.* (SEC 15), 2015, pp. 913–928.
4. Z. Xu, H. Wang, and Z. Wu, "A Measurement Study on Co-residence Threat inside the Cloud," *Proc. 24th USENIX Security Symp.* (SEC 15), 2015, pp. 929–944.
5. Y. Zhang et al., "Cross-VM Side Channels and Their Use to Extract Private Keys," *Proc. 19th ACM Conf. Computer and Communications Security* (CCS 12), 2012, pp. 305–316.
6. Y. Zhang et al., "Cross-Tenant Side-Channel Attacks in PaaS Clouds," *Proc. 21st ACM Conf. Computer and Communications Security* (CCS 14), 2014, pp. 990–1003.
7. V. Varadarajan et al., "Resource-Freeing Attacks: Improve Your Cloud Performance (at Your Neighbor's Expense)," *Proc. ACM Conf. Computer and Communications Security* (CCS 12), 2012, pp. 281–292.
8. M. Walfish and A.J. Blumberg, "Verifying Computations without Reexecuting Them," *Comm. ACM*, vol. 58, no. 2, 2015, pp. 74–84.
9. F. Armknecht et al., "A Guide to Fully Homomorphic Encryption," Cryptology ePrint Archive, report 2015/1192, 2015; eprint.iacr.org/2015/1192.pdf.
10. A. Everspaugh et al., "The Pythia PRF Service," *Proc. 24th USENIX Security Symp.* (SEC 15), 2015, pp. 547–562.
11. A. Juels and R.L. Rivest, "Honeywords: Making Password-Cracking Detectable," *Proc. ACM Conf. Computer and Communications Security* (CCS 13), 2013, pp. 145–160.
12. S.K. Fayaz et al., "Bohatei: Flexible and Elastic DDoS Defense," *Proc. 24th USENIX Security Symp.* (SEC 15), 2015, pp. 817–823.
13. A. Chi et al., "Server-Side Verification of Client Behavior in Cryptographic Protocols," *Proc. 14th USENIX Symp. Networked Systems Design and Implementation* (USENIX NSDI 17), 2017.
14. V. Varadarajan, T. Ristenpart, and M. Swift, "Scheduler-Based Defenses against Cross-VM Side-Channels," *Proc. 23rd USENIX Security Symp.* (USENIX Security 14), 2014.
15. S.-J. Moon, V. Sekar, and M.K. Reiter, "Nomad: Mitigating Arbitrary Cloud Side Channels via Provider-Assisted Migration," *Proc. 22nd ACM Conf. Computer and Communications Security* (CCS 15), 2015, pp. 1595–1606.
16. S.K. Fayazbakhsh et al., "Enforcing Network-Side Policies in the Presence of Dynamic Middlebox Actions Using Flowtags," *Proc. 11th USENIX Symp. Networked Systems Design and Implementation* (NSDI 14), 2014, pp. 533–546.
17. A. Gember et al., "OpenNF: Enabling Innovation in Network Function Control," *Proc. ACM Conf. Special Interest Group on Data Communication* (SIGCOMM 14), 2014, pp. 163–174.
18. C. Prakash et al., "PGA: Using Graphs to Express and Automatically Reconcile Network Policies," *Proc. ACM Conf. Special Interest Group on Data Communication* (SIGCOMM 15), 2015, pp. 29–42.
19. Y. Zhai et al., "CQSTR: Securing Cross-Tenant Applications with Cloud Containers," *Proc. ACM Symp. Cloud Computing* (SoCC 16), 2016, pp. 223–236.
20. F. Pouget, M. Dacier, and H. Debar, *Honeypot, Honeynet, Honeytoken: Terminological Issues*, white paper RR-03-081, Institut Eurecom, Sept. 2003.

Jay Aikat is a research associate professor in the Department of Computer Science at the University of North Carolina at Chapel Hill (UNC-CH). She's also the chief operating officer (COO) at the Renaissance Computing Institute (RENCI). Her research interests are in computer networking and cloud security. Aikat received a PhD in computer science from UNC-CH. She's a member of ACM. Contact her at aikat@cs.unc.edu.

Aditya Akella is an associate professor in the Department of Computer Sciences at the University of Wisconsin—Madison. He received a PhD from Carnegie Mellon University. Contact him at akella@cs.wisc.edu.

Jeffrey S. Chase is a professor of computer science at Duke University. His research interests include infrastructure control and trust management for networked services. Chase received a PhD in computer science from the University of Washington. He's a member of ACM and USENIX. Contact him at chase@cs.duke.edu.

Ari Juels is a professor at the Jacobs Institute at Cornell Tech. His current research interests include applied cryptography, blockchains and smart contracts, cloud security, and the use of deception in computer security. Juels received a PhD in computer science from the

University of California, Berkeley. He's a member of ACM. Contact him at juels@cornell.edu.

Michael K. Reiter is the Lawrence M. Slifkin Distinguished Professor in the Department of Computer Science at UNC-CH. His research interests include computer security, distributed computing, and networking. Reiter received a PhD in computer Science from Cornell University. He's a Fellow of ACM and IEEE. Contact him at reiter@cs.unc.edu.

Thomas Ristenpart is an associate professor at Cornell Tech. His research interests include security, privacy, and cryptography. Ristenpart received a PhD in computer science from the University of California, San Diego. Contact him at ristenpart@cornell.edu.

Vyas Sekar is an assistant professor in the Electrical and Computer Engineering Department at Carnegie Mellon University. His research interests lie at the intersection of networking, security, and systems. Sekar received a PhD in computer science from Carnegie Mellon University. Contact him at vsekar@andrew.cmu.edu.

Michael Swift is an associate professor in the Department of Computer Sciences at the University of Wisconsin—Madison. Swift received a PhD from the University of Washington. Contact him at swift@cs.wisc.edu.