SANS | Research Program

2026 EDITION

Architecting the Agentic Cloud:

Building Secure AI Agents with Azure

Written by:
• Wesley Kuzma
• Chris Cochran

In partnership with:

Microsoft

August 2026

About the Authors

Wesley Kuzma,
Microsoft, Senior Director Security Architecture

Wesley Kuzma has over 17 years of experience in cybersecurity and currently serves as the Sr. Director Security Architecture within Microsoft's Engineering & Architecture Group. He is a frequent presenter to diverse audiences, including Fortune 500 executives, engineering teams, and internal Microsoft stakeholders. Wesley holds multiple advanced degrees and serves as an Adjunct Professor at several universities across the United States.

Chris Cochran
SANS Institute, Field CISO and Vice President of AI Security

Chris Cochran is the Field CISO and Vice President of AI Security at the SANS Institute, where he stands at the intersection of frontier AI innovation and real-world cybersecurity defense.

A Marine Corps veteran and former leader at organizations like Netflix, Mandiant, the U.S. House of Representatives, Axonius, and NSA, Chris has spent his career translating complexity into clarity, whether guiding organizations through emerging AI threat landscapes, pioneering defensive AI workflows, or shaping the next generation of AI-security practitioners. His work blends deep operational cyber experience with cutting-edge research in AI governance, model risk, multiagent systems, and adversarial AI, positioning him as one of the few leaders equally fluent in shaping AI strategy and securing it.

Known for his storytelling, community leadership, and ability to distill fast-moving technical shifts into actionable insights, Chris is helping build the future where AI systems are both powerful and safe.

Introduction: Why Agentic AI Changes Everything About Enterprise Security

We are at an inflection point that comes once in a generation. For decades, enterprise software did what it was told. It processed transactions, surfaced dashboards, and responded to queries. It was deterministic. Predictable. Auditable. The security model we built, encompassing perimeter defense, identity control, and data classification, was designed for a world where humans made decisions and machines executed them. That world is ending.

Agentic AI systems reason, plan, and act. They hold memory across sessions, invoke tools autonomously, delegate tasks to other agents, and operate at machine speed inside environments that were built for human-paced decisions. A single agent deployment can touch email, databases, code repositories, and financial systems in a single workflow, making dozens of decisions without a human in the loop.

It is a structural shift in the relationship between software and organizational authority.

The urgency is real and accelerating. Gartner forecasts that by 2028, 33% of enterprise software applications will use agentic AI, enabling 15% of day-to-day work decisions to be made autonomously. Organizations that fail to establish governance frameworks before agentic systems reach production are accumulating security and compliance liability at a pace that outstrips their ability to remediate it.

This chapter is written for the security and cloud teams who must govern this shift. It addresses the pro-code agentic ecosystem on Microsoft Azure, where the power is highest, the risk surface is broadest, and the need for architectural discipline is most acute. Citizen development and low-code tooling, while important, are deliberately out of scope. The enterprise-grade, programmatic deployment model is where the largest decisions, and the largest exposures, currently live.

The central argument is this: Secure, governed deployment of agentic AI is a foundational architectural imperative.

Core Agentic Security Primitives

With the agentic evolution upon us, we are proposing six core primitives to help organizations adapt as technology continues to rapidly move into the agentic space. Together, these core primitives provide the identity, data, posture, access, runtime, and audit controls needed to adopt AI agents safely (see Figure 1). Keep in mind these core primitives are vendor-agnostic and should be considered during any agentic implementation regardless of who owns the platform.

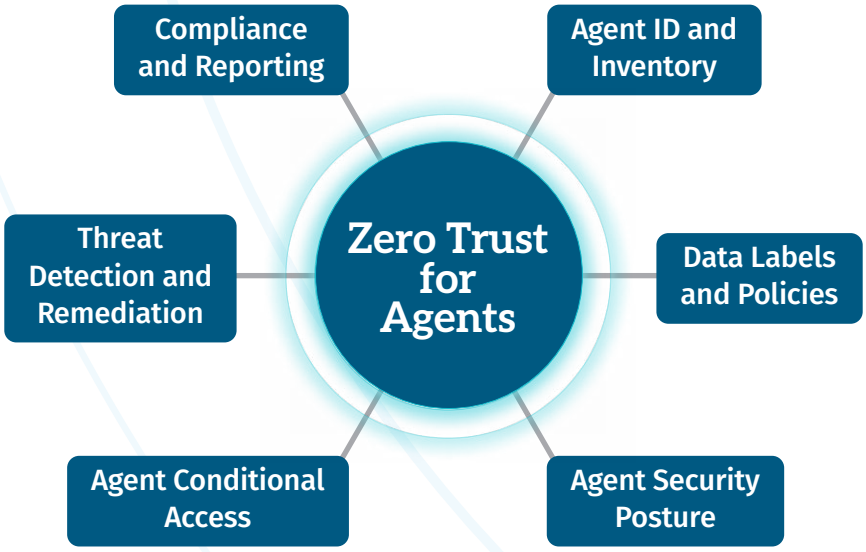

Figure 1. Six Core Primitives

Agent Identification and Inventory

You cannot govern what you cannot see. The identity control plane must be extended into your agentic implementation. This is similar to the approach the industry has taken with traditional software as a service (SaaS) applications. When your network no longer can provide a boundary, you are only left with identity as your boundary, which is arguably even more important in agentic applications than it has been in SaaS applications. The identity plane must provide each agent with a first-class directory object with an owner, life cycle, and audit trail. Every agent should be given an agent ID at creation and registered with a centralized tracking mechanism.

These concepts also should extend to identifying, tracking, and governing "shadow agents." For the purposes of this paper, let's consider shadow agents as professional, production-use agents not developed through your organization's approved professional development processes. Your agentic inventory tooling must support multifaceted identification of these shadow agents. It is not enough to just look at network traffic or running processes or agentic authentication activity. You must take a comprehensive, layered approach to identifying shadow agents.

Data Labels and Policies

Data governance remains foundational heading into 2026. Agents are only as trustworthy as the data they can reach. Labels and policies can help lower the risk of agentic solutions in two ways:

1. Labeled and encrypted content should honor the user's authorization before it can be returned by an agent. To get to this level of maturity, though, your sensitive data must be labelled as a precursor to professional agentic development activities. Your solution should have an authorization capability that respects data labels and specifies data usage.

2. Data labels and policies also can be applied when end users are interacting with an agentic solution. This includes developers using AI tools to develop new agentic solutions. For example, data loss prevention policies should be in place that evaluate and potentially block the upload of sensitive data types to generative AI tooling.

Microsoft Recommendation

Start by labeling the M365 data stores (SharePoint, OneDrive) that contain the most sensitive information in the organization. This is a good way to start to reduce data risk and show tangible progress, even in a large organization.

Agent Security Posture

Agents are not singular entities. An agentic solution is made up of many entities' models, datasets, libraries, tools, services, and skills—and, yes, even the agent code itself can be treated as a separate entity within the overall agentic solution. Each component of an agentic security solution will have specific configurations that may adversely impact the solution's security posture. Just as we have with cloud services, we need tooling that automatically identifies these misconfigurations, categorizes them by severity, and offers remediation assistance (see Figure 2).

Figure 2. Agent Posture

Agent Conditional Access

Identity by itself is not access. We must make mandatory conditions-based logic part of every authentication event. This logic should reinforce the principles of Zero Trust and allow authentication to be tied to distinct attributes like network location, compliance signals, and even correlated risk signals. Every agentic authentication event should be tied to this logic.

We also should be able to tie conditions-based logic to humans invoking agentic solutions. This looks closer to traditional conditional access logic and is tied to signals like MFA status, compliance, and geolocation.

Microsoft Recommendation

If using Entra ID for conditional access, your current rules should be re-evaluated to ensure they take into account agentic solutions. A number of recent features have been added that allow you to better target and tune conditional access policies to agentic solutions.

Agentic Threat Detection and Remediation

Once agents are deployed, runtime threats must be detected and contained. Example threats include: prompt injection, indirect injection through grounding data, credential abuse, sensitive-data exfiltration, and risky tool use. You must be able to source signals from your agentic solutions that provide timely insights into potential agentic-specific threats to the right people in your organization (see Figure 3).

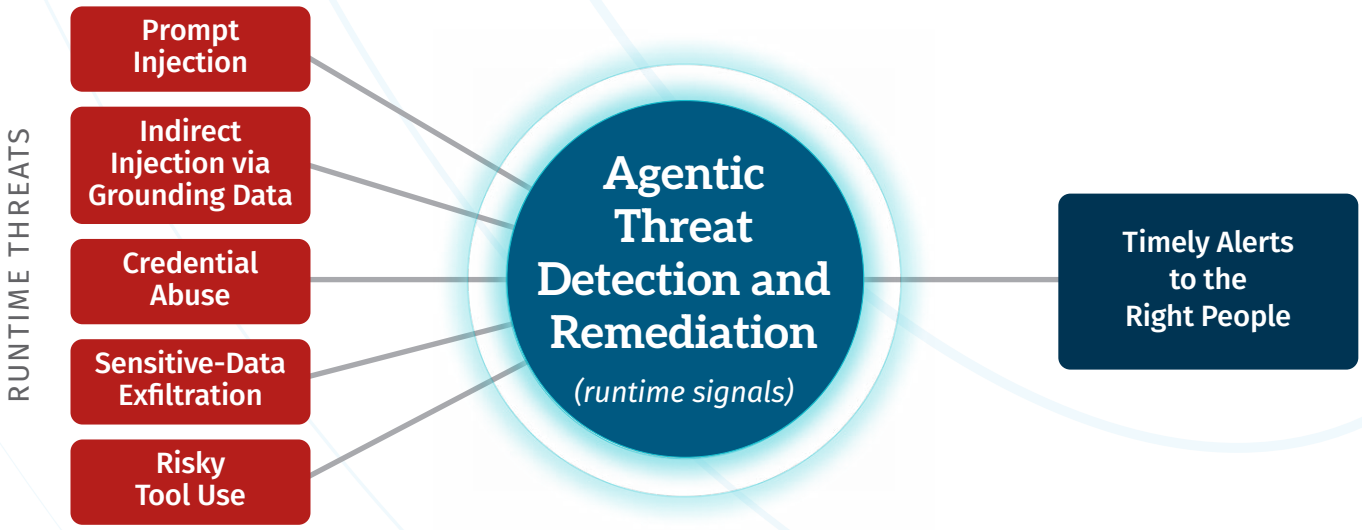

Figure 3. Agentic Threat Detection and Remediation

Compliance and Reporting

Regulators are attempting to move as fast as technology. In an effort to get ahead of the technical innovation, government bodies around the world are drafting new regulatory guidelines for agentic solution operations. Templated regulatory guidance should be available as part of your solution that will automatically assess the solution against evolving regulatory frameworks and policies such as EU AI Act or NIST AI RMF.

Microsoft Recommendation

Start by aligning your AI-specific policy enforcement with existing known requirements. A good example of this would be to align your prompt retention policy with your existing email retention standard.

In conclusion, these six primitives should not just be viewed as new products bolted onto AI. They should be a deliberate extension of your Zero Trust architecture to a new class of nonhuman entity.

Agentic Capabilities in Azure

Microsoft Azure has made a substantial, coordinated investment in becoming the enterprise platform of choice for agentic AI. This is not a collection of siloed AI experiments. It is a coherent stack designed for organizations that need to move from pilot to production at scale. Microsoft Foundry is the key professional AI development service hosted in Azure. Understanding Foundry and the supporting stack of services is the first step toward understanding how to secure and govern AI development and services in the Azure Cloud.

The strategic significance of Azure's agentic platform is that it compresses what previously took a custom engineering team months to build—orchestration, tool-calling, memory management, and observability—into managed services with enterprise service level agreements (SLAs). This dramatically lowers the barrier to deployment. The same dynamics that made cloud adoption a security-forcing function a decade ago are now at work with agentic AI: The business will move fast, and security must move with it or be bypassed.

For CISOs and cloud security architects, the operative question is not "Should we allow agentic systems?" That decision has likely already been made in your organization. The operative question is "Can our existing cloud security controls, built for human-initiated, stateless interactions, adequately govern systems that are stateful, autonomous, and capable of lateral tool invocation?"

The honest answer, for most organizations today, is no. The gap has little to do with the platform's controls. Security teams have not yet mapped their threat models, access governance, and monitoring architectures to the new shapes that agentic workloads create.

This section establishes the Azure services that form the foundation of enterprise agentic deployment and can be used to enforce the agentic security primitives. The goal is to give security and governance teams a precise understanding of the attack surface they are inheriting and the native controls available to reduce it.

Multiagent Architecture and MCP Server

As previously mentioned, a professional agentic solution is complex. Many individual components make up the overall solution, and the majority of professionally developed agentic solutions leverage a multiagent architecture that also may call numerous other tools, APIs, skills, and services. Enterprise AI platforms are increasingly building security and governance capabilities that account for this complexity, and these architectures can be used to support many of the security primitives and control overall solution risk. For example, the security and governance capabilities being built into Microsoft's AI platform capabilities take into account these complex agentic architectures.

Facilitating a multiagent architecture requires an orchestration layer, which in many cases is an agent itself. This is often referred to as an "orchestration agent," which could be thought of as the "brains of the agentic solution." This orchestration agent can delegate security and governance tasks to purpose-built agents as part of the development process. These purpose-built agents should carry out tasks like code scanning, compliance checking, and risk scoring. Platforms are beginning to expose this orchestration directly. For example, the Microsoft Agent Framework represents these workflows both visually and as YAML-authorable files.

If the orchestrator agent is the "brain," then the Model Context Protocol (MCP) server is the "nerves" that connect signals from the brain to the rest of the solution.

Azure illustrates one way this pattern is being implemented. In Foundry Agent Service, an agent connects to one or more remote MCP servers, either public endpoints or private endpoints hosted within a dedicated subnet, and gains access to their tools without bespoke integration code. Authentication is centralized rather than embedded in agent code, and governance is enforced at the point of each tool call, including allow-listing which tools an agent may invoke and requiring human approval for high-risk operations. The pattern generalizes: Any platform supporting MCP should centralize authentication and gate tool access by default rather than trust agents to self-limit.

The practical guidance holds regardless of platform. Treat MCP servers as governed supply-chain components. Security teams should prefer first-party catalog entries over arbitrary third-party servers, host sensitive servers privately with internal-only ingress, scope each connection with least-privilege tokens, require approval for any tool that writes data, and log every tool invocation for audit. When combined with the workflow patterns above, this produces a multiagent architecture that is modular by design, governable by default, and portable across runtimes, Microsoft's included.

If the orchestrator agent is the 'brain,' then the Model Context Protocol (MCP) server is the 'nerves' that connect signals from the brain to the rest of the solution.

Identity for Agents: The Three-Body Problem of Enterprise Identity

For years, security and IAM teams have been playing catch-up on two fronts simultaneously. Human workforce identities (employees, contractors, and partners) multiplied across SaaS applications, cloud environments, and hybrid infrastructure, each with their own life cycle, entitlements, and access patterns. Alongside them, nonhuman identities (NHIs) stealthily became the larger problem: Service accounts, API keys, OAuth tokens, machine credentials, and automation pipelines proliferated faster than any governance framework could track. Most enterprises today have five to 10 times as many NHIs as human identities. Most of those NHIs are under-governed, over-privileged, and poorly audited.

In classical mechanics, the two-body problem, predicting how two objects with mass interact under gravity, has a clean, elegant solution. Add a third body, and the math breaks down. The system becomes chaotic, sensitive to initial conditions, and analytically unsolvable. This is precisely what happens to enterprise identity when agentic AI enters the picture. Workforce identities, legacy NHIs, and agentic identities interact across dynamic, multilayered environments in ways that are not just more complex; they are categorically different. Determining who is doing what, where, why, and on whose authority at any given moment becomes, quite literally, a three-body problem. The interactions are nonlinear. The state changes constantly. And the consequences of a single miscalculation propagate at machine speed.

> **Workforce identity was hard. Nonhuman identity made it harder. Agentic identity makes it something else entirely.**

Agents don't work shifts. They don't have cognitive limits. They can be spawned dynamically, operate across dozens of tools in a single workflow, delegate tasks to subagents, and act on behalf of human principals, without those principals being aware in real time. An agent provisioned for a narrow task can, if improperly scoped, inherit access to systems far beyond its operational mandate. And unlike a human employee who performs unauthorized access once and leaves a trail, an agent can act continuously, systematically, and at a scale that overwhelms manual audit review.

NHIs, such as machine accounts, service identities, and agent-based API keys, play a key role in agentic AI security, with agents often operating under NHIs when interfacing with cloud services. Yet the governance frameworks most enterprises have built for NHIs were designed around static, predictable workloads (e.g., a service account that calls one API, on a known schedule, from a known network location). Agentic identities violate every one of those assumptions. They are dynamic. They are contextual. They reason about their environment and adapt their behavior. And they can be manipulated, through prompt injection, memory poisoning, or goal hijacking into taking actions the governing identity was never intended to authorize.

The result is a new class of identity risk that sits at the intersection of all three bodies: a human principal delegates authority to an agent operating under an NHI, which then coordinates with other agents under different identities, across systems where no single team has full visibility. When something goes wrong—and in insufficiently governed environments, something will—attribution is not just difficult. It may be impossible without purpose-built instrumentation designed for this architecture from the start.

Identity governance was already hard. Agentic AI changes the physics of the problem entirely.

The strategic mandate is therefore both more urgent and more specific than traditional NHI governance. It is not sufficient to have identities for agents. Every agent must have an identity that is scoped to least privilege for its specific task, time-bound to its operational window, continuously monitored for behavioral deviation, and revocable without cascading disruption to dependent workflows. Every action taken under that identity must be attributable, not just to the agent, but to the human principal and business process that authorized it. Agentic systems introduce new failure modes, including tool misuse, prompt injection, and data leakage, and identity is the control plane that determines which of those failure modes are contained and which become enterprise-wide incidents.

Identity is the foundational prerequisite for every other control discussed in this paper. An agent with a well-governed identity operating in a poorly governed environment is a partial defense. An agent with a poorly governed identity operating anywhere is an open door.

Azure's Entra ID, Managed Identities, and ABAC/RBAC architecture provide the native building blocks to establish this foundation at enterprise scale. The critical question, and where this section will go deep, is how Microsoft is evolving these capabilities specifically for the agentic use case: workload identity federation for dynamic agent contexts, emerging patterns for delegated agent authority that preserve human accountability, and the operational practices that turn identity governance from a provisioning checklist into a runtime security function capable of keeping pace with the three-body problem it now has to solve.

Data Security for AI

The data-security challenge for agentic systems is not new in kind, but it is new in volume and shape. Agents read and write across more data sources than a traditional application, often produce derivative content, and frequently operate without a human in the immediate loop.

Purview, Microsoft's data policy and posture layer, illustrates what this looks like in practice:

- **Data security posture management (DSPM) for AI** allows for sensitivity-label propagation through AI-generated outputs and endpoint data loss prevention rules that block sensitive content from being pasted into unsanctioned consumer AI services such as ChatGPT and Gemini.
- **Insider risk management** adds a dedicated "Risky AI usage" policy template that detects prompt-injection attempts and other anomalous interactions with generative AI .
- **eDiscovery** now indexes Copilot interactions, and Compliance Manager ships AI-specific regulatory templates aligned to emerging frameworks including the EU AI Act.

These capabilities are continuing to converge into a single AI governance surface rather than a collection of disconnected tools.

Platform Governance Tools

Enterprise AI platforms increasingly provide a resource hierarchy and configuration surface purpose-built for governing agent behavior: tenant-level isolation, RBAC that can be delegated at the project level without granting tenant-wide privileges, and centralized policy for encryption keys, network configuration, and content safety. Azure's Foundry is one example of this pattern; the specific console varies by platform, but the underlying governance surface is now fairly consistent across vendors.

These controls typically include inline guardrails that evaluate model inputs and outputs against configured harm categories, plus additional filtering aimed at indirect prompt injection, where retrieved external content attempts to override an agent's instructions. For high-assurance environments, centralized guardrail and policy management across an organization's fleet of agents becomes a baseline requirement, not an advanced feature.

Microsoft Recommendation

1 Label first, then build.

Sensitivity labels applied at the source propagate through Copilot and Foundry-grounded retrieval, allowing downstream agent outputs to inherit the most restrictive label of any contributing source.

2 Constrain retrieval, not just generation.

The most reliable way to keep an agent from leaking data is to ensure its grounding index cannot retrieve data the requesting principal could not see directly. Foundry knowledge indexes honor caller identity through OBO when configured correctly.

3 Treat synthetic outputs as production data.

Agent-generated artifacts inherit the label and risk profile of their inputs. Storage and exfiltration controls must therefore apply to agent outputs as well as inputs.

AI Is Both the Problem and the Solution: Finding the Holes Before the Adversary Does

While enterprises were still debating governance frameworks and drafting AI acceptable-use policies, threat actors were already operationalizing AI to accelerate reconnaissance, generate more convincing phishing campaigns, automate vulnerability discovery, and craft prompt injection payloads tailored to specific enterprise deployments. The evolution of AI-based threat actors is already reshaping the cybersecurity landscape in real and measurable ways. The asymmetry is stark: Attackers benefit from AI's ability to compress the time between identifying a weakness and exploiting it, while defenders are still largely relying on human-paced processes to detect, investigate, and respond. Closing that gap requires more than better tools. It requires a fundamental shift in how security teams think about their own velocity.

The answer is not solely to fight AI with legacy processes. It is also to fight AI with AI, and the most powerful expression of that principle is the convergence of purple teaming with AI-driven adversary emulation. Purple teaming, at its core, is the disciplined integration of offensive and defensive security functions: red team techniques that expose real weaknesses, blue team capabilities that detect and respond, and the continuous feedback loop between them that produces an iteratively hardening security posture. When AI is applied to both sides of that equation—generating novel attack scenarios at a pace and level of creativity that no human red team can match, while simultaneously accelerating the detection, triage, and remediation of the gaps those scenarios expose—purple teaming transforms from a periodic exercise into a continuous, self-improving security function. The environment gets harder to compromise at the same cadence at which the threat landscape itself evolves.

For agentic AI deployments specifically, this capability is existential. Hidden prompts have already turned enterprise copilots into silent exfiltration engines, agents have bent legitimate tools into destructive outputs, and leaked credentials have allowed them to operate far beyond their intended scope. These are failure modes already documented from the first generation of production agentic deployments, and they represent only the attacks that have been discovered and disclosed. The attacks that haven't been found yet are the ones that matter most. AI-driven adversary emulation, systematically probing agent orchestration layers, MCP server configurations, identity delegation chains, and memory contexts for exploitable weaknesses, gives security teams the only realistic mechanism for discovering those vulnerabilities on their own terms, before an adversary does. The goal is an agentic security environment that improves faster than it can be compromised: an almost organic defense that learns from every simulated attack and closes every identified gap before it becomes an incident.

Conclusion

Agentic AI is a structural change in who and what holds authority inside the business. Agents reason, plan, and act at machine speed across identities, data, tools, and other agents, and they will do so whether or not security teams have caught up. The argument of this paper is that catching up is not a matter of buying more controls. It is a matter of architecting them deliberately, in the right order, around the realities of how agentic systems actually behave.

The six core primitives (Agent ID and Inventory, Data Labels and Policies, Agent Security Posture, Agent Conditional Access, Threat Detection and Remediation, and Compliance and Reporting) provide the vendor-agnostic spine. They are the deliberate extension of Zero Trust to a new class of nonhuman actor, and they sequence the work for security leaders who must move from pilot to production without accumulating unbounded liability: identify, label, harden, gate, detect, and report.

Taken together, these layers describe an agentic cloud that is modular by design, governable by default, and defensible at machine speed. The organizations that will adopt agentic AI safely are not the ones with the most policies. They are the ones that treat secure agent deployment as a foundational architectural imperative, embedding identity, data, posture, access, runtime, and audit controls into the platform from day one rather than retrofitting them after the first incident. The technology will continue to accelerate. The window to architect ahead of it is open now, and it is the defining cloud security decision of this generation.

SANS | Research Program

About the SANS Research Program

The SANS Research Program is a key initiative by the SANS Institute and a premier global provider of cybersecurity research and information. SANS Research Program is designed to provide cybersecurity practitioners and leaders with data-driven insights, thought leadership, and solutions that help them better understand and respond to evolving security challenges. All content is authored by SANS instructor experts from around the world who apply their years of experience from hands-on practitioner work in the field, advisory roles, and the classroom to provide education, guidance, and actionable insights that help make the cyber world a safer place.

To learn about sponsorship opportunities for research, content, and in-person or virtual events, email us at **Sponsorships@sans.org** or go to **www.sans.org/sponsorship**.

Free Resources

Access more free educational content from SANS at:

SANS Reading Room
www.sans.org/white-papers

Check out upcoming and on-demand webcasts:

X
x.com/SANSInstitute

LinkedIn
www.linkedin.com/company/sans-institute

YouTube
www.youtube.com/@SANSInstitute