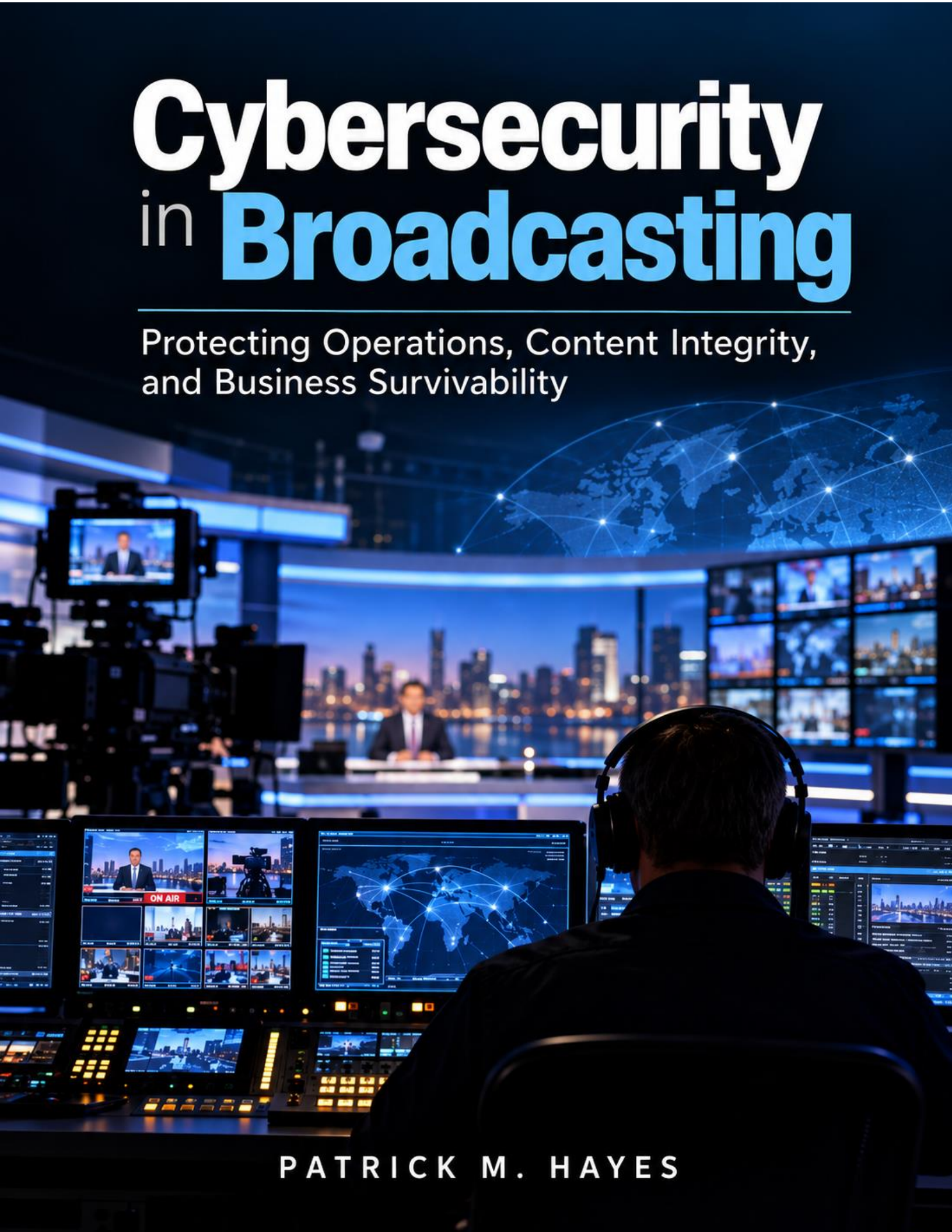
Cybersecurity in Broadcasting

Protecting Operations, Content Integrity, and Business Survivability



PATRICK M. HAYES

© 2026 Integrated Assurance LLC. All rights reserved. Integrated Assurance® is a registered trademark.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means without the prior written permission of the author, except for brief quotations used in reviews, commentary, or scholarly work.

For permission requests, contact Integrated Assurance LLC at info@integratedassurance.co.

Table of Contents

Executive Summary 4

Broadcasting Has Become a Digital Dependency Business 5

The Broadcast Attack Surface Is Now an Operating Environment 8

Identity Is Now Part of the Broadcast Control Plane 9

Ransomware, Extortion, and Operational Disruption 11

Cloud Production and the New Concentration Risk 12

Third Parties Are Part of the Broadcast Environment 14

AI Is Becoming Operational Infrastructure 16

Agentic AI Changes the Meaning of Access 18

Content Integrity in an Era of Synthetic Media 21

Live Production Creates a Different Risk Profile 22

Newsrooms and Editorial Operations Need Their Own Security Model 24

Where Physical and Cyber Risk Meet 26

Incident Response Must Be Built Around Decisions 27

Business Continuity Must Reflect How Broadcasting Actually Works 29

Recovery Has to Restore the Business 31

Security Operations Need Broadcast Context 32

Governance Has to Reach the Decisions That Create Risk 34

Cyber Insurance Is Part of Financial Resilience 36

Measure What Tells You Something 38

Integrated Assurance as the Operating Model 40

90-Day Agenda for Broadcast Leadership 42

Conclusion: Protecting the Ability to Broadcast 43

Executive Summary

Broadcasting has changed considerably over the last decade, and cybersecurity has not always kept up with that change. We still tend to think about broadcast security in terms of protecting networks, production systems, transmission infrastructure, and the content itself. Those things obviously still matter, but they represent only part of the environment that broadcasters now depend on.

Consider what it takes to operate a broadcast organization today. Production may happen in the cloud. Employees and contractors work remotely. Vendors routinely connect directly to systems they support. Newsrooms exchange content through third-party platforms. Identity services control access to applications that may have little connection to one another beyond the fact that they trust the same credentials. Streaming and digital distribution have added another collection of providers and technologies. AI is now being introduced into many of these workflows, sometimes with the knowledge of the organization and sometimes without it. Very little of this is unusual anymore. That is precisely why it matters.

The cybersecurity problem has become much bigger than protecting the broadcast network. A broadcaster can have a well-protected network and still have a serious operational problem if Microsoft, AWS, a production vendor, an identity provider, or another critical service becomes unavailable. The same is true if an employee account is compromised and provides access to several systems at once. None of these situations necessarily involves someone attacking the broadcast signal. They can still prevent the organization from doing its job.

That changes where the cybersecurity conversation should start. The broadcast operation itself is a better starting point than a list of threats or controls.

Instead of starting with threats and controls, start with the broadcast operation. What absolutely has to work for the organization to produce and distribute content? What does the newsroom depend on? What happens if the normal production environment is unavailable? Which vendors have access to critical systems? How much of the operation depends on one identity provider? If a cloud service goes down, is there actually another way to work or does the recovery plan simply assume the service will come back? Those are cybersecurity questions now.

AI adds another dimension to this because it is quickly becoming part of the normal technology environment. Broadcasters are already using AI for transcription, research, editing, translation, advertising, audience analysis, scheduling, and production support. There is nothing inherently alarming about that. Most organizations will probably use considerably more AI over the next several years because there are plenty of legitimate reasons to do so. The concern is what happens as these tools gain access to more of the business.

There is a fairly significant difference between asking an AI tool to summarize a transcript and allowing an AI agent to access a content repository, interact with other applications, change information, or initiate an action. In the second example, we are no longer talking only about whether the AI produces an accurate answer. We are giving software authority to do something.

That raises some fairly basic questions. What can it access? What can it change? What information is it allowed to use? Does anyone know when it takes an action? Can that action be reversed? Most importantly, who is responsible for the outcome?

I do not think broadcasters need an entirely new philosophy of cybersecurity to answer these questions. They do need a much better understanding of how technology is being used across the organization and where authority is being delegated to technology.

Synthetic media presents a related problem. There has been a tremendous amount of attention given to deepfakes and whether technology can reliably detect them. Detection is useful, but I would not build a content integrity strategy around the assumption that we will always be able to identify a convincing fake.

The more practical issue for a broadcaster is whether it can establish where content came from and why it trusts it. Who provided it? Has it been changed? Who reviewed it? Who approved it for publication? If someone challenges its authenticity after it goes on air, what does the organization have to support its decision?

These are not entirely new questions for journalists. What has changed is the speed and quality with which false material can now be created. That puts considerably more pressure on processes that may already be operating under tight deadlines.

Meanwhile, the threats we have been dealing with for years have not gone away. Ransomware can still shut down an organization. Stolen credentials still provide attackers with access. Poorly secured remote access still creates opportunity. Vendors still get compromised. The difference is that these events can now move through a much more interconnected environment.

This is especially important in broadcasting because time is unforgiving. If a manufacturer loses access to a business system for several hours, there may be ways to absorb the delay. Broadcasting does not always have that luxury. The newscast still starts at six. The game still begins at seven. Breaking news does not wait for the identity provider to come back online. That is why I believe business survivability is a better way to frame the cybersecurity discussion.

No broadcaster can prevent every disruption. A more useful objective is to understand what the business cannot afford to lose and make reasonable decisions about how those capabilities will continue when something goes wrong.

The answer will vary. A security control or additional redundancy may be appropriate in one case, while another problem may be better handled through a manual process, another vendor, a different communications method, insurance, or clearer authority to make a decision during an incident.

The question I explore is if an important part of the broadcast environment fails tomorrow, can the organization still broadcast? Answering that question requires looking well beyond the security department. It requires understanding how the business actually works, what it has become dependent on, and whether the organization has enough evidence to believe its plans will work when it needs them.

Broadcasting Has Become a Digital Dependency Business

For much of broadcasting's history, resilience was easier to see. The most important systems were specialized, usually located in facilities the broadcaster controlled, and supported by engineering teams that understood how everything fit together. Transmission equipment had a known purpose. Production systems had clear owners. The path from the studio to the audience could be traced through infrastructure that was largely designed, operated, and maintained by the broadcaster. When redundancy was needed, it could usually be built around equipment, power, facilities, transmission paths, and other things the organization understood well.

Broadcasting does not work that way anymore. The traditional broadcast infrastructure is still there, but it now operates alongside a much larger collection of digital services. Newsrooms depend on software platforms that may be hosted somewhere else. Production teams move content through cloud services. Employees, contractors, and outside contributors connect remotely. Live production can involve public cloud infrastructure and third-party platforms. Content moves through traditional broadcast channels while also being distributed

through streaming services, websites, mobile applications, social media, and other digital platforms. Advertising, analytics, audience measurement, and content management introduce even more technology into an environment that was already complicated.

Over time, this has changed what broadcasters actually depend on to operate. A company may own its studios, production equipment, and transmission infrastructure while still being dependent on an outside identity provider just to give employees access to the systems they need. DNS may be provided by another company. Content delivery may rely on several external services. Telecommunications providers connect facilities and remote users. Software vendors control licensing and support. Managed service providers may have administrative access to parts of the environment. Collaboration platforms have become so embedded in daily operations that losing access to them can create an immediate communication problem during an incident.

Some of these dependencies are easy to see because everyone knows they exist. Others have developed gradually and receive very little attention until something goes wrong. A production team adopts a cloud service because it solves a problem. A vendor is given remote access because it makes support easier. Two systems are connected through an API because it eliminates a manual process. A collaboration platform becomes the place where people coordinate nearly everything they do. None of these decisions necessarily looks significant at the time, but together they can change how the business operates and what it can no longer operate without.

This is where traditional views of cybersecurity can become limiting. Security teams are accustomed to thinking about assets, vulnerabilities, threats, and controls. Those things still matter, but knowing that a system is secure does not tell us how important that system has become to the business. It also does not tell us what happens when the system is unavailable for reasons that have nothing to do with an attack.

A cloud outage can create the same operational problem as a cyber incident if the affected service is critical to production. A compromised administrator account can reach several parts of the company when those systems rely on the same identity platform. A vendor incident can interrupt production even though none of the broadcaster's own technology was directly attacked. What starts as a technology problem can become a broadcast problem very quickly.

That is why broadcasters need to understand their environment in terms of dependency, not simply ownership. The fact that the company does not own a particular service does not make that service any less important to its operation. In some cases, the systems the broadcaster controls least may be the ones it depends on most.

Looking at the business this way also changes how we think about what is critical. Producing a live newscast is not a system. Neither is putting a scheduled channel on air, publishing breaking news online, delivering advertising, or communicating with the public during an emergency. These are things the business needs to be able to do. Technology supports them, but so do people, vendors, data, facilities, communications, and decisions made by individuals across the organization.

If leadership starts with those activities, it becomes much easier to understand where the real exposure exists. A live newscast may depend on newsroom systems, production technology, identity, network connectivity, external feeds, cloud services, graphics, communications, and people with very specific skills. The individual components may look resilient when they are reviewed separately. The weakness may not become apparent until someone looks at what they all have in common.

Identity is a good example. Several systems may have their own infrastructure and separate vendors but still rely on the same identity provider. From a technology perspective, the systems appear independent. From an operational perspective, they may share a single point of failure. If employees cannot authenticate, the fact that the applications themselves are still running may not matter very much.

The same problem can exist with cloud infrastructure. Two vendors may appear to provide completely different services while both depend on the same underlying cloud provider or region. A broadcaster may believe it has diversified its technology when the underlying dependency has simply moved somewhere it cannot easily see. The same can happen with telecommunications, content delivery, security services, and other parts of the digital supply chain.

Business continuity plans can carry similar assumptions. An organization may have a detailed recovery plan that depends on employees communicating through Microsoft Teams, Slack, email, or another collaboration platform. That works until the identity or cloud problem being managed also affects the platform everyone planned to use during the recovery. The plan exists, but one of its basic assumptions was never tested.

These are not necessarily failures of planning or poor technology decisions. Most of them develop because organizations make reasonable decisions over time without having a good way to see how those decisions change the business as a whole. Each system has an owner. Each vendor has a contract. Each application may have gone through some form of review. What is often missing is the ability to see the dependency that has developed across all of them.

Cloud adoption makes this more difficult because technology can be introduced much faster than it could when new infrastructure required equipment, facilities, and a formal implementation project. A production team can adopt a software service quickly and have it become an important part of the workflow long before anyone thinks to include it in continuity planning. An API can connect two systems and quietly create a dependency that may not be obvious to either system owner. The technology works well enough that nobody has much reason to think about the relationship until one side becomes unavailable.

This is also why a traditional asset inventory only tells part of the story. Knowing that the organization has an application, server, cloud service, or production system is useful. Knowing what the business uses it for is much more useful. If a system becomes unavailable, leadership should be able to understand what stops working, how quickly that becomes a problem, and whether there is another reasonable way to perform the work.

Data should be viewed the same way. Broadcasters hold enormous amounts of information, but not all of it has the same operational importance at the same moment. An archive may be extremely valuable to the organization while having little effect on tonight's broadcast if it becomes temporarily unavailable. Losing access to the current rundown five minutes before air is a very different problem. Losing the credentials needed to control a distribution platform may be more urgent still.

That does not mean one type of information deserves protection and another does not. It means protection and recovery should reflect how the information is actually used. Data classification can help, but it becomes much more useful when it is connected to the work the broadcaster is trying to perform.

None of this requires the organization to create a perfect map of every technology relationship. That would probably be outdated before it was finished. The more practical approach is to begin with the parts of the business that matter most and work backward. Understand what is required to keep those activities running, where the organization is heavily dependent on one provider or technology, and where a failure would leave few realistic alternatives.

Once those dependencies become visible, leadership can make much better decisions about them. Some may justify additional redundancy. Others may need a tested manual workaround. A vendor relationship may need more attention because the organization has become more dependent on it than anyone realized. In some cases, leadership may decide that the concentration is acceptable because the cost of eliminating it is greater than the risk. That is still a legitimate decision when it is understood and made deliberately.

For leadership, the practical issue is how these dependencies affect the ability to broadcast and what options remain when something important is unavailable. That puts cybersecurity much closer to the actual operation of the business, where many of the consequences are now felt.

The Broadcast Attack Surface Is Now an Operating Environment

When we talk about attack surface, we tend to think about the systems and services an attacker can reach. That is still useful, but it does not really describe what has happened to the broadcast environment. There are simply too many things connected to it now, and many of them are outside the traditional production network.

A typical broadcaster is running the same business technology as almost any other company while also supporting newsroom systems, media management, editing, graphics, automation, playout, encoders, storage, archives, streaming, transmission and a collection of specialized production equipment. Some of it may be relatively new. Some of it may have been in place for years because it works and there has been no good reason to replace it.

That last point matters in broadcasting. An older production system is not necessarily sitting there because somebody forgot about it. It may be there because it is stable, the vendor still supports it, and changing it creates a risk to production. Anyone who has worked around operational technology understands the problem. Security may want the system patched or upgraded. Engineering may know that doing so could affect something that needs to be on air tonight.

I would much rather see that risk understood than force the system into a corporate security standard that does not fit. If endpoint protection cannot run on a piece of production equipment, then we need to look at what else can be done around it. Maybe it should be isolated. Maybe there are only two accounts that should ever administer it. Maybe the vendor should only be able to connect when someone approves the session. Maybe we need better monitoring around it. There are usually options, but they depend on understanding what the system does and how it is actually used.

This is one of the places where broadcast engineering and cybersecurity have to work together. The security team knows something about the threats and the ways an attacker may get into the environment. Engineering knows what can be changed without disrupting production. Neither team has the complete answer on its own.

Remote production has made that relationship more important. People who used to work almost entirely from a studio or production facility can now work from home, from a hotel, from a venue or from just about anywhere they can get a reliable connection. Equipment moves around with them. Contractors come and go. Vendors can work on systems remotely instead of sending someone onsite.

Most broadcasters could not operate the way they do today without some of that flexibility. Trying to solve the security problem by forcing everyone back behind a traditional network boundary is not realistic. The better question is how much confidence we have in the person connecting, the device being used and the access they have once they get there.

The public-facing environment has grown just as quickly. A broadcaster may have websites, streaming platforms, mobile applications, cloud storage, APIs, content portals and remote management services spread across several providers. Different groups may be responsible for them. Some may have been built internally. Others may have come through a vendor or been added for a specific project.

This is where fairly simple mistakes can become important. A storage bucket gets exposed. An administrative page ends up reachable from the internet. Someone puts a credential in code. A service that was supposed to

be temporary never gets shut down. These are not particularly exotic security problems, but they are easy to create when the environment is changing all the time.

Advertising is part of this environment as well. Digital advertising has its own platforms, data, analytics, scripts and outside providers. It may sit well outside the systems the broadcast engineers think about every day, but that does not make it unimportant. If an incident disrupts advertising or exposes audience information, the station can lose revenue or create a privacy problem without anything happening to the broadcast itself.

The same applies to the rest of the business. Payroll and finance do not put a program on air. Legal does not run playout. Microsoft Teams or Slack may have nothing to do with the signal. They can still become very important in the middle of an incident.

Think about a ransomware event that hits finance while the company is trying to recover production. Somebody may need to authorize an emergency purchase or pay a vendor and suddenly the normal process is unavailable. Or consider an incident where everyone planned to coordinate through the same collaboration platform that has just gone down. Those systems may not be part of broadcasting in the traditional sense, but the company has become dependent on them to operate.

That is why I do not think a modern broadcast environment can be understood from a network diagram alone. Security sees part of it. Engineering sees another part. Production knows where the workarounds are. The business knows which failures start costing money. The people responding to an incident need pieces of all of that information if they are going to understand what is actually happening.

There will always be gaps. Some production equipment will not provide the kind of logging we would like. Some systems will be difficult to patch. There will be vendors that need unusual access because of the equipment they support. There will be temporary production environments that have to be stood up quickly. Pretending those conditions do not exist does not improve security.

What matters is whether we know about them and have made reasonable decisions around them. If a system cannot be monitored directly, perhaps we monitor the network around it. If it cannot be patched, perhaps we reduce the number of systems that can reach it. If a vendor needs administrative access, we should at least know when they are using it and why.

This is also where I think security programs can become too focused on proving that a control exists. Saying a production network is segmented is useful. Knowing that the segmentation has actually been tested is better. Saying vendor access is controlled sounds good until someone discovers that three old vendor accounts are still active. Having a recovery plan matters, but I would put much more confidence in the organization that has actually tried to use it.

Broadcasting will continue to adopt new technology because it has to. Cloud production, remote workflows, automation and AI are not going away because they make the security environment harder to manage. The challenge is to keep enough visibility into how the environment is changing that security does not fall several steps behind the business.

That requires less emphasis on making every system look the same and more attention to understanding what is actually there, how it is being used and what happens to the broadcast operation when something goes wrong.

Identity Is Now Part of the Broadcast Control Plane

Identity has become one of the most important control surfaces in broadcasting because so many systems now trust the same user accounts, authentication services, and administrative roles. The practical perimeter of the

organization is often the point where a person or service proves who it is and receives permission to act. If that process is compromised, an attacker may not need to defeat a network boundary at all.

The risk is larger than stolen passwords. Modern attacks target sessions, tokens, help desk processes, multifactor enrollment, privileged accounts, application credentials, service accounts, and the relationships between identity providers and cloud services. A user can have strong multifactor authentication and still be exposed if an attacker steals an authenticated session or convinces support staff to reset an account. Identity security therefore needs to account for how access is granted, maintained, elevated, recovered, and revoked.

Broadcast environments make this harder because access needs can be unusual. Freelancers, temporary production staff, outside producers, engineering vendors, contractors, talent, and remote crews may need access for a short period. Major events can create sudden staffing changes. Vendors may need emergency access at inconvenient hours. Production deadlines can encourage teams to keep old accounts active or share credentials because doing so feels operationally easier than following a slower process.

Those workarounds accumulate risk. An account created for a contractor may remain active after the engagement ends. A vendor account may have broad access because no one wants to discover during an outage that the vendor cannot reach the system it supports. Service accounts may use credentials that rarely change because nobody is certain what will break. Privileged roles may be assigned permanently even though the user needs them only occasionally.

The better model is to treat identity as operational infrastructure. Access should reflect the work being performed and should change when that work changes. Privileged access should be temporary where practical. Administrative actions should be attributable to an individual or controlled service. Emergency access should be designed in advance rather than improvised during a crisis. Offboarding should include contractors, vendors, applications, and machine identities, not only employees.

This becomes especially important in cloud production. A privileged cloud identity may be able to alter storage, networking, compute resources, logging, or security controls across a large environment. The account may have more operational power than a traditional administrator had inside a single facility. Strong authentication is necessary, but it should be accompanied by limits on privilege, separation of duties where it matters, monitoring of administrative activity, and recovery procedures for the identity platform itself.

Identity also affects content integrity. Publishing systems, social accounts, streaming platforms, content management systems, and distribution portals rely on trusted identities to establish who is authorized to release material. A compromised account can create a false appearance of legitimacy. The organization therefore needs a way to distinguish ordinary user access from authority to publish, approve, alter, or remove content that represents the broadcaster publicly.

Service identities and AI agents add another challenge. Software increasingly acts on behalf of users. An automation may move content, modify metadata, call an external service, create a summary, publish a file, or trigger another workflow. Agentic AI can expand that pattern by selecting actions dynamically. The security question becomes whether the organization can identify the acting service, understand the permissions it has, and determine which human remains accountable for what it does.

Identity recovery deserves the same attention as identity protection. If the primary identity provider is unavailable, the broadcaster needs to know which critical systems remain accessible and how authorized staff will gain access. If privileged accounts are suspected of compromise, teams need a way to establish clean administrative access without relying on the same accounts being investigated. These scenarios should be exercised because the first time an organization discovers that all recovery paths depend on one identity system should not be during an incident.

When identity is treated as part of the broadcast control plane, security decisions become clearer. The organization can focus on the identities that can materially change operations, content, or security posture. It can reduce persistent privilege, tighten vendor access, improve lifecycle management, and build recovery paths around the accounts that matter most. This is more useful than treating identity as an isolated authentication project because it connects access directly to operational consequence.

Ransomware, Extortion, and Operational Disruption

Ransomware is particularly difficult for broadcasters because so much of the business runs against a clock. There is a point when the newscast starts, the game goes live, programming airs, or breaking news needs to be published. An incident does not change that schedule. The technical team may still be trying to determine what happened while the production team is already dealing with what needs to happen next.

This is also why I think the impact of ransomware is sometimes described too narrowly. Encryption is certainly a problem, but it may not be the problem that creates the most disruption. An attacker may have stolen credentials, gained administrative access, moved through several systems, copied information, damaged backups, or compromised identity before anyone knew they were there. Once that is discovered, the organization has to make decisions about systems that may still be running but can no longer be completely trusted.

That can put a broadcaster in a difficult position. Taking a system offline may interrupt production, while leaving it online may allow the attacker to maintain access or move farther into the environment. There may not be enough information yet to know which choice is right. The people making that decision need to understand both sides of it because simply choosing the most conservative security response can have real consequences for the broadcast.

I would want to know well before an incident what we absolutely need to keep operating. That does not mean every system that someone has labeled critical. It means understanding what is really required to put something on air or publish information when the normal environment is not available. For one station, that might mean keeping a basic on-air capability running without the normal newsroom systems. Somewhere else, it could mean having another way to receive content or publish important information when the primary platform cannot be used.

The answer does not always have to be another technology platform. There may be a manual process that is perfectly adequate for several hours. There may be equipment that can be operated locally if remote management is unavailable. A different communications method may be enough to keep the right people coordinated. What matters is that someone has thought through how the work would actually get done rather than assuming the recovery team will figure it out at the time.

I have the same concern with backups. Organizations can spend a lot of money on backup technology and still not know how long it would take to recover the business. Restoring the data does not necessarily restore the application that uses it, the identity service needed to access it, the network connection needed to reach it, or the configuration needed to make everything work together again. There may also be a vendor involved, and that vendor may not be able to respond as quickly as the recovery plan assumes.

The only reliable way to learn some of this is to actually try it. Pick something important and restore it. Find out who needs to be involved and what information they need. See where credentials come from and whether the documentation is current. Find out whether the license still works. Determine whether somebody has to call the vendor. A recovery exercise tends to expose small assumptions that would otherwise stay hidden, and

several small assumptions can consume a surprising amount of time when the organization is already under pressure.

The sequence matters as well. Infrastructure teams have good reasons for restoring technology in a particular order, but the business may need something different first. If the immediate concern is getting the next newscast on air, that should influence what gets attention. The recovery plan has to account for the way the broadcaster actually operates, not just the technical relationships between servers and applications.

There is another part of ransomware that is easy to underestimate until it happens. While the technical teams are investigating and recovering systems, executives may already be dealing with an extortion demand, questions from employees, calls from business partners, legal concerns, insurance requirements, and the possibility that stolen information will be released. They may be making those decisions before anyone can give them a confident explanation of what happened.

That is a bad time to start figuring out who is supposed to be involved. Legal counsel should already be known. The cyber insurer should already be part of the incident process. Leadership should know who can authorize outside forensic help and who can speak for the company. The organization should also have a way to reach these people that does not depend entirely on corporate email or the collaboration system that may be caught up in the incident.

Cyber insurance can help, especially when the policy provides access to experienced breach counsel, forensic firms, negotiators, and other resources that the broadcaster does not maintain internally. I would still want to understand how that process works before needing it. Knowing who to call, what the carrier expects, and whether outside providers need to be approved is fairly basic preparation, but it is much easier to sort out during normal business than during an extortion event.

This is also why I prefer ransomware exercises that leave some things unanswered. Real incidents rarely arrive with a clean description of what the attacker did. It is more likely that an administrator account appears to be compromised, several systems are behaving strangely, and someone has found a ransom note. Nobody knows yet whether data was stolen or whether the attacker reached production. That is enough information to create a problem, but not enough to make the decisions easy.

Put the leadership team in that situation and the discussion becomes much more useful. They have to decide whether a system that is still working should remain online. Production may want to keep using something that security cannot yet say is safe. Communications may want to make a statement before the investigation is complete. Legal may have concerns about what can be said. Meanwhile, the next broadcast is getting closer.

That is much closer to the problem a broadcaster is likely to face. Ransomware resilience is not really demonstrated by having a response plan or even by having good backups. It becomes apparent when the organization has to keep operating without some of the technology and information it normally takes for granted. A broadcaster that has already worked through those decisions has a much better chance of keeping a difficult incident from becoming a much larger operational crisis.

Cloud Production and the New Concentration Risk

Cloud technology has given broadcasters options that simply did not exist when most production infrastructure had to live inside a facility. Capacity can be added quickly, production can happen from different locations, and content can move between people and platforms without having to build everything around physical infrastructure. For a business that regularly deals with changing production demands, that flexibility has real value. It also means that a surprising amount of the operation can end up depending on a relatively small number of providers.

That dependency is not always obvious. A broadcaster may have dozens of cloud applications and assume that diversity provides some protection, even though many of those applications ultimately run on the same cloud provider. Several may also depend on the same identity service or content delivery network. If that shared service has a significant problem, applications that appeared unrelated can become unavailable at the same time.

I do not see this as an argument against cloud production. In many cases, cloud technology can make the organization more resilient. The issue is knowing where the dependencies are and making sure the business understands them before an outage or security incident exposes them.

There is also still some confusion about what the cloud provider is responsible for and what remains with the customer. Moving a system into AWS, Azure, Google Cloud, or a software-as-a-service platform does not move every security responsibility with it. The provider may be responsible for the underlying infrastructure while the broadcaster is still responsible for who has access, how the service is configured, what happens to the data, and how the organization recovers if something goes wrong.

This becomes more important because cloud environments can change very quickly. That is one of their biggest advantages, but it is also where mistakes can spread. A configuration can be copied across dozens of resources. Infrastructure as code can make deployments much more consistent while just as consistently deploying a bad setting. APIs make it possible to automate work that once required someone to log into a system and make a change manually, which also makes the credentials behind those APIs extremely important.

Logging is another example. Most major cloud platforms are capable of producing an enormous amount of information about what is happening in the environment. That does not necessarily mean the information will be there when someone needs it. The right logging still has to be enabled. It needs to be retained long enough to be useful and protected from someone who may be trying to cover their tracks. Someone also needs to know where to find it during an investigation.

The pressure to move quickly can make some of this harder. Broadcast production does not always operate on the timetable that a formal technology governance process would prefer. A new environment may be needed for an event next week. A production partner needs access today. Someone needs a quick way to exchange a large amount of content. A test environment works well enough that people begin using it for real work.

Those are normal business decisions, and I would be cautious about solving them by adding more approvals. If every cloud change requires a long security review, people will eventually find ways around the process because they still have a job to do. It is usually more practical to give teams a safe way to move quickly.

This is where standard cloud designs can help. If the organization already has an approved way to handle identity, administrative access, logging, encryption, networking, and backups, a production team does not have to reinvent those things every time it creates something new. Automated checks can catch some of the obvious mistakes without waiting for someone in security to manually review every deployment. The production team can still move quickly, but it is doing so inside an environment where some of the important security decisions have already been made.

The harder question is what happens when the cloud provider itself has a problem. A service can be designed to be highly available and still depend on a single region, management service, identity platform, or network path. If that shared dependency fails, the redundancy inside the application may not help very much.

The obvious response is to add more redundancy, but that can become expensive and complicated very quickly. Running in multiple regions may make sense for an important service. Running the same workload

across completely different cloud providers may sound even safer, but now the organization has two environments to operate, secure, test, and recover. If the team does not have the people or experience to manage that additional complexity, the architecture intended to improve resilience can create problems of its own.

I would make those decisions based on what the workload actually does for the broadcaster. If a service can be unavailable for several hours without materially affecting the operation, building an elaborate alternate environment may not make sense. If the service is necessary to keep a live channel running, publish emergency information, or support an important production, the answer may be very different.

There may also be a simpler fallback. The organization might be able to operate manually for a period of time, move to a reduced service, or use another distribution path. That may provide enough resilience without trying to duplicate an entire cloud environment. The important thing is that the decision reflects what the business needs rather than a general belief that more redundancy is always better.

Broadcasters should also think about how difficult it would be to leave a cloud service. That does not mean maintaining a second provider just in case the relationship ends. It means understanding what would actually have to move. Data is usually the obvious part, but configurations, integrations, operating procedures, specialized knowledge, and the way people have built their daily work around the platform can be much harder to replace.

A service that was easy to adopt can become very difficult to leave after several years of integrations and operational changes. That may be an acceptable business decision, but leadership should understand when it has happened. Otherwise, what looked like a convenient technology choice can quietly become a dependency the organization has very few practical options to replace.

The supply chain behind cloud services makes this even harder to see. A broadcaster may buy a production application from one company without having a direct relationship with the cloud provider hosting it, the identity service it uses, the content delivery network supporting it, or some of the software embedded in the product. A problem with any one of those providers can still affect the broadcaster.

This is why I would not look at cloud risk simply by counting cloud vendors or asking whether a provider has the right security certifications. I would want to understand what happens to the broadcast operation if an important cloud service is no longer available and how much control the organization still has when that happens.

Cloud technology can absolutely make broadcasting more flexible and more resilient. It can also allow dependencies to build quietly because the infrastructure behind the service is no longer visible in the same way it was when everything lived in a broadcast facility. The important part is knowing where those dependencies have developed and making deliberate decisions about the ones that could materially affect the broadcaster’s ability to operate.

Third Parties Are Part of the Broadcast Environment

Broadcasting has always relied on outside companies. Equipment manufacturers, telecommunications providers, production companies and other specialists have been part of the business for decades. What has changed is how far many of those companies now reach into the broadcaster's day-to-day operation.

A vendor may host an application used by the newsroom, remotely administer production equipment, process or store content, manage a security platform, support streaming, provide advertising technology, or maintain infrastructure that the broadcaster cannot easily operate without them. In some cases, the vendor has privileged

access directly into the environment. In others, the broadcaster may have very little control over the technology at all because the entire service is operated by the provider.

That makes the usual approach to third-party risk feel incomplete. Vendor questionnaires, security reviews and contract requirements all have a place, but I would want to know something much more basic about an important provider. What happens to us if they are not there tomorrow?

The answer can be very different depending on the vendor. Losing a provider that supplies a useful business application may be inconvenient. Losing the company that supports a critical production platform a few hours before a major event could create a much bigger problem. The amount being spent with the vendor does not necessarily tell us anything about how important the relationship has become.

This is why I would look at vendor criticality based on what the broadcaster depends on the company to do. A relatively inexpensive service can become deeply embedded in a production workflow. Something that started as a convenience may now be the only practical way a team performs a particular task. That dependency may never appear in a traditional vendor review because nothing about the size of the contract suggests that the provider is particularly important.

Remote access deserves the same kind of attention. Vendors often need access to the systems they support, especially with specialized broadcast technology where much of the expertise remains with the manufacturer or service provider. There is nothing inherently wrong with that arrangement, but the broadcaster should know who has access, what they can reach, and whether that access still reflects what the vendor actually needs.

This can get messy over time. A vendor account created during an installation remains active because the vendor may need it again. Another account is added for a support engineer. Someone creates a shared account because it is easier than managing individual access. Years later, the relationship is still active, but nobody is entirely sure who at the vendor can use those credentials.

I would be particularly careful with permanent administrative access into production. If a vendor only needs access occasionally, there is little reason for that connection to remain open all the time. When access is needed, it should be possible to identify who is using it and what they are doing. That becomes important during an incident because the broadcaster may need to determine very quickly whether activity came from its own people, a vendor, or someone using a vendor's credentials.

Contracts can help, although there is a limit to what contract language can accomplish during an actual incident. I would certainly want requirements around security and data handling, but I would also want to understand what the vendor is expected to do when something goes wrong. How quickly are they required to tell us about an incident? Who provides support? What information will they share? What happens to our data when the relationship ends? How is remote access removed?

Those questions become much more important when the vendor is part of an essential operation. If a production provider has an incident during a major broadcast, the broadcaster may need information immediately. Finding out that the contract gives the provider several days to notify customers may not be particularly helpful when decisions have to be made in the next hour.

There is another issue that can be harder to see because it may sit behind the vendor relationship. Two providers can look completely independent and still depend on the same infrastructure. They may run on the same cloud provider, use the same identity service, depend on the same telecommunications carrier, or incorporate the same software into their products.

That becomes important when the organization believes those vendors provide redundancy. Moving from one service to another does not help very much if the same underlying outage takes both of them down. The

broadcaster does not need to understand every supplier behind every vendor, but for services that are genuinely important, it is worth understanding where the obvious shared dependencies exist.

A vendor incident creates another practical problem because the broadcaster may have to make decisions before the provider knows exactly what happened. The provider may say it is investigating while the broadcaster is trying to decide whether to disconnect remote access, stop sending data, move production somewhere else, or warn people internally not to use the service.

Waiting for complete information is not always an option. There should be some understanding of when the broadcaster will act on its own, particularly when a vendor has privileged access or supports an important production capability. That does not mean disconnecting every vendor at the first sign of trouble. It means having enough understanding of the relationship to make a reasonable decision while the facts are still developing.

I would also want to know which systems the broadcaster cannot realistically support without the vendor. This is fairly common with specialized technology. The internal team may operate the system every day but still depend on the manufacturer for major troubleshooting, software recovery, licensing, or replacement components.

That may be perfectly reasonable. Maintaining deep internal expertise for every specialized platform would be expensive and probably unnecessary. The problem is assuming that the vendor will always be available when needed. If losing vendor support would make an important system impossible to recover, that should at least be understood before the organization has to deal with it.

There may be simple ways to reduce some of that dependency. Better documentation may be enough in one case. Another system may justify spare equipment or some additional internal training. For something especially important, the organization may want to understand whether another company could provide support if the primary vendor could not. The answer will depend on the system and how long the broadcaster can reasonably operate without it.

This is the part of third-party risk that I find more useful than trying to assign every vendor a risk score. I want to know what we depend on the vendor for, what they can access, what information they hold, and what happens to the operation if they have a serious problem. I also want to know whether there is another way to perform the work or whether the business has effectively committed itself to waiting for the vendor to recover.

None of this changes the fact that broadcasters need specialized partners. There is no practical reason to bring every capability inside the company, and doing so would not necessarily make the organization more resilient. The important thing is knowing when a useful vendor relationship has become an important business dependency and making sure the broadcaster still has some ability to respond when that relationship does not work as expected.

AI Is Becoming Operational Infrastructure

AI is already part of the way many broadcasters work. It is being used to transcribe interviews, search archives, translate material, create captions, summarize information, assist with editing, support research, analyze audiences and speed up some of the routine work that goes into producing content. Advertising and marketing teams are using it as well. In many cases, employees started using these tools long before the organization had a formal discussion about AI.

That is probably the most practical place to start with AI governance. Before deciding what the policy should say, find out what people are actually doing.

That can be harder than it sounds because AI does not always arrive through a formal technology project. An employee can open a public AI service in a browser and start using it immediately. Someone else may install an extension or connect an application using a personal account. Existing software can add AI features through a routine product update. A platform that went through a security review two years ago may now be capable of doing things that were never considered when it was approved.

Trying to solve this by blocking everything that has not been formally approved is unlikely to work for very long. People are using these tools because they are useful. If the approved process is too difficult or the company does not provide a reasonable alternative, some of that use will simply become harder to see.

I would start by understanding where AI has become part of the work. What are people using it for? What information are they giving it? Does the output stay with the employee, or does it eventually become part of something the broadcaster publishes or relies on? Can the AI only generate an answer, or has it been connected to other systems? Those differences matter much more than whether a particular product happens to be called AI.

There is not much reason to treat an employee using AI to help draft an internal email the same way we would treat a system that can publish content or change information in a production platform. The first use may require little more than some reasonable rules about what information can be entered into the service. The second deserves much more attention because the technology has been given some authority to affect the business.

This is where I would separate using AI for assistance from giving AI the ability to act. A journalist asking for help summarizing a long document is still making the decision about what to do with the result. If an AI system can change production metadata, publish material, send information outside the company, approve an advertisement or initiate some other action, the situation is different. Someone needs to understand exactly what the system has permission to do and how an unwanted action can be stopped or reversed.

Data is one of the more immediate concerns because it is very easy to give an AI system information that would never intentionally be sent to an outside party. Someone working quickly may paste in an unpublished story, a contract, internal correspondence or information about a confidential source without giving much thought to what happens after pressing enter. They may not know whether the provider retains that information, uses it to improve a model, or makes it available through some other feature of the service.

A policy can address some of this, but employees need guidance they can actually use. Most people are not going to stop in the middle of their work and read several pages of AI policy before deciding whether they can summarize a document. They need to understand which tools the company is comfortable with and what kinds of information should never be entered into them. For sensitive work, the rules should be especially clear.

The output deserves some common sense as well. Generative AI can produce something that reads extremely well and is still wrong. That becomes more important in broadcasting because a polished answer can move quickly from research into a script, caption, translation, graphic or published story. The problem is not that AI makes mistakes. People make mistakes too. The problem is how easily a confident answer can be accepted because it looks finished.

The amount of review should depend on what happens if the AI gets it wrong. Using it to organize notes for an internal meeting is one thing. Using it to translate a statement that will be broadcast is something else. Using it as part of research for a breaking story may require a different level of verification again. Existing editorial judgment still applies, even if the technology producing the first draft is new.

The vendors behind these tools need some attention too. AI products are changing much faster than most traditional business software. A service approved today may have substantially different capabilities six months from now. It may add connectors to email, files, calendars or collaboration platforms. A feature that originally required someone to copy and paste information may later be able to retrieve that information directly.

That means approval cannot always be treated as permanent. The question is not whether the vendor passed a review once. The broadcaster needs to notice when the product gains capabilities that change the original risk decision, particularly when those changes involve access to company information or the ability to act in other systems.

There is also an editorial question around disclosure. Broadcasters will have to decide when audiences should be told that AI was involved in creating or modifying content. I do not think every use of AI automatically requires a label. Using AI to clean up a transcript is very different from creating a synthetic image of an event that never occurred. Those decisions belong with editorial and legal leadership, but they are difficult to make consistently if nobody knows where AI is being used in the first place.

Much of AI governance can therefore fit into things the broadcaster should already be doing. Access still needs an owner. Sensitive information still needs to be protected. Vendors still need to be understood. Important content still needs to be verified. Someone still has to be responsible for a decision, even when software helped make it.

The technology will continue to change, and AI will increasingly disappear into products people already use every day. At some point, asking whether the organization "uses AI" may not be a particularly useful question because the answer will almost always be yes. What will matter is what the technology has been allowed to do inside the business, what it can access, and whether the organization still knows who is responsible for the result.

Agentic AI Changes the Meaning of Access

Agentic AI changes the conversation because the technology is no longer limited to generating an answer for someone to review. An agent can be connected to other systems and given the ability to do things. It might retrieve information, update a record, create a file, send a message, change a schedule, or start another process without someone manually performing each step.

There are plenty of places where that could be useful in broadcasting. Production involves a lot of coordination between people and systems, and some of that work is repetitive. An agent that can gather information from several places and complete routine tasks could save time. The concern begins when we give the agent enough access to make decisions or changes that matter to the operation.

I think access is where this becomes much more interesting from a security perspective. We have spent years getting better at managing what employees can access, although most organizations would probably admit there is still plenty of work to do. Now we are beginning to create another class of identity that may have access to many of the same systems without being a person at all.

An agent needs credentials or some other form of delegated access to interact with those systems. Depending on how it is configured, that access can be fairly narrow or surprisingly broad. The problem is that the permissions are often established when the integration is created, while the way people use the agent continues to evolve afterward. Something originally connected to retrieve information may eventually be used for much more.

This is why I would be careful about treating an AI agent like another employee. The interface may feel conversational and the system may appear to understand what someone is asking, but that does not mean it understands the business in the way a person does. It is still software operating with whatever instructions, information, and permissions it has been given. The fact that it communicates naturally can make it easier to forget that distinction.

I would pay much more attention to what the agent is technically capable of doing. If it only needs to read a production schedule, there is no reason to give it permission to change the schedule. If it is helping someone draft social media content, publishing that content should probably remain a separate decision unless there is a good reason to automate it. If an agent is working with archive information, access to unrelated legal, personnel, or financial information should not come along simply because everything happens to live on the same platform.

This sounds like ordinary least privilege because that is largely what it is. The difference is that access management programs have traditionally been built around people, groups, and applications. Agents add another layer of machine identities, API tokens, service accounts, application permissions, and delegated access that someone needs to keep track of.

There will also be situations where it makes sense to let an agent prepare an action without letting it complete the action on its own. An agent could assemble the information, recommend the change, and present it to someone for approval. For something that could affect live production, published content, money, or sensitive information, that extra step may be worth the few seconds it adds.

The approval itself needs to mean something. A button that simply asks someone to approve an action without showing what is about to happen does not provide much protection. The person should be able to see what the agent intends to change and have enough information to decide whether it makes sense.

Logging becomes particularly important once agents are taking actions across several systems. If something goes wrong, I would want to know what the agent was asked to do, what information it used, where it went, and what it changed. I would also want to know which credentials allowed the action to occur. Without that information, an investigation can quickly turn into people trying to reconstruct a chain of automated activity from several unrelated platforms.

There is another complication because agents do not necessarily receive instructions only from the person using them. They may also process documents, websites, emails, messages, or other information as part of completing a task. Some of that content can contain instructions intended to manipulate the agent. This is generally described as prompt injection, and it becomes much more important when the agent has access to information or systems beyond the content it is currently reviewing.

An agent that can only summarize a document has limited room to cause damage if it is manipulated. An agent that can read the same document and then access email, cloud storage, or production systems creates a very different situation. The amount of trust placed in the agent should reflect what is on the other side of its credentials.

Not every problem will involve an attacker. An agent can simply get something wrong. It can misunderstand a request, use incomplete information, repeat a task, or make a change that is technically allowed but makes no sense operationally. Anyone who has worked with automation has seen some version of this problem before. AI makes the automation more flexible, but it does not eliminate the possibility of a bad outcome.

For that reason, I would want a straightforward way to shut an agent down. Its credentials should be identifiable and revocable. People should know who owns it and who has the authority to disable it. If the

agent changes something important, there should be some thought given to how that change can be undone. These are much easier questions to answer before the agent becomes part of a critical workflow.

Broadcasting will probably find many good uses for this technology. Scheduling, advertising, archive management, content preparation, audience engagement and production coordination all involve work that could benefit from more capable automation. Technical operations may eventually use agents as well. I do not think the answer is to hold those uses back until every possible risk has been solved.

I do think broadcasters should be careful about allowing useful automation to quietly become operational authority. An agent that helps someone do a job is one thing. An agent that can act across several systems and affect production is something else. As that authority grows, the broadcaster needs to remain clear about what the agent can reach, what it is allowed to change, and who is responsible for it.

That is probably the most durable way to approach agentic AI because the technology itself will change quickly. The specific products and capabilities available a year from now will almost certainly be different from what organizations are experimenting with today. The basic management problem will remain much more familiar. Someone has been given access to important systems, except this time that someone is software.

Content Integrity in an Era of Synthetic Media

Broadcasters have dealt with questionable source material for a long time. A video arrives from someone nobody knows. A clip starts circulating on social media without much context. A source makes a claim that cannot immediately be confirmed. None of that is new. What AI changes is how easy it has become to create something convincing enough to get through an initial review.

That matters most when there is pressure to move quickly. During breaking news, there may not be much time between receiving something and deciding whether it is usable. A video can look completely believable and still be fabricated. The same is now true of a phone call or voice message. Even photographs have become harder to judge by appearance alone. The old instinct that something looks wrong is not going to be enough in every case.

There are products that try to detect whether media has been generated or manipulated. I would use them as another source of information, but I would not want a newsroom making an editorial decision based entirely on what one of those tools says. The content may have been edited or compressed several times before it reaches the newsroom. It may have been downloaded from one platform and uploaded to another. The technology creating synthetic media is changing too quickly to assume a detector will always recognize what it is looking at.

In practice, I think the source of the material becomes more important. If a video comes from a journalist standing at the scene, there is already some context around it. If the same video appears through an unknown social media account, there is much less to work with. A newsroom may need to contact the person who posted it, find another view of the event, check details in the background, or see whether someone it trusts can confirm what happened. That is familiar journalistic work even though the reason for doing it has changed.

There are efforts underway to make the history of digital content easier to establish. Content credentials and similar approaches can preserve information about where a file came from and what happened to it after it was created. I think that will become useful, particularly when cameras, editing software, and publishing platforms support it more consistently. It is unlikely to solve the entire problem. A lot of legitimate news material will continue to arrive without that information, especially material coming from the public.

The newsroom still has to make a judgment in those cases. Sometimes there will be enough information to use the material confidently. Sometimes there will not. The difficult cases are the ones in between, especially when competitors are already publishing the content and the pressure to move becomes much harder to ignore.

Synthetic media also creates problems that have nothing to do with what appears on air. Someone can use a cloned voice to impersonate an executive or a producer. A reporter could receive a message that appears to come from a source they already know. Finance could get a call that sounds like someone in senior management asking for an urgent payment. These are ordinary fraud techniques made considerably more convincing by better technology.

Employees need some way to deal with requests that do not fit the normal pattern. Calling someone back on a known number may be enough in one situation. A financial transaction may require confirmation through an established process. A newsroom dealing with sensitive source material may use an entirely different method. I would not try to create one verification procedure for all of these situations because the work is too different.

There is also a problem that runs in the opposite direction. Real audio or video can now be challenged simply by calling it AI-generated. A person does not have to prove that something is fake to create doubt around it. In some situations, raising the possibility may be enough to confuse an audience or delay a story.

That gives broadcasters another reason to keep track of important source material and how it was handled. If a piece of reporting is challenged later, the newsroom may need to go back to the original file, the source, the reporter's notes, or other material used during verification. That record may end up being far more useful than trying to prove authenticity with a detection tool after the story has already become controversial.

Broadcasters are also going to create synthetic material themselves. AI-generated translation is an obvious example. So are synthetic voices, images, graphics, and other production elements. Some uses may be routine enough that audiences barely notice them. Others could leave an audience with a very different impression of what actually happened.

Those decisions are going to require editorial judgment. A translated version of an interview is not the same as generating video of a person doing something they never did. There is a lot of territory between those examples, and broadcasters will have to decide what they are comfortable producing and when an audience should be told how something was created.

Mistakes will happen. Eventually, a broadcaster will publish something that turns out to have been manipulated, just as news organizations have published incorrect information from other sources. What matters at that point is whether the organization can figure out what happened quickly enough to respond.

That can become complicated once the material has spread. Removing the original post does not remove the copies, screenshots, clips, and commentary that may already be circulating. Editorial and communications teams may be dealing with the public response while security or technical teams are trying to determine where the material came from. Legal may have its own concerns. That is not the time for those groups to be figuring out for the first time who is responsible for what.

I do not think broadcasters need to build an entirely new content integrity program around AI. Much of what they already know about sources, verification, editorial review, corrections, and reputation still applies. The difference is that convincing false material is becoming much easier to produce, while authentic material is becoming easier to challenge. That puts more weight on the broadcaster being able to explain why it trusted something in the first place and what it knows about where that material came from.

Live Production Creates a Different Risk Profile

Live production does not leave much room for a long discussion about what to do next. If something starts behaving strangely ten minutes before a newscast or in the middle of a live event, the people running the production have to make a decision. They may not know whether they are dealing with a security incident, an equipment problem, a bad configuration, or something else entirely.

That uncertainty is part of what makes live production different. Taking a system offline may be the safest security decision and still be the wrong operational decision at that moment. Keeping it online may preserve the broadcast while allowing a problem to get worse. Someone has to make that call with whatever information is available at the time.

People who work in these environments already make tradeoffs to keep production stable. An engineer may decide not to install an update before an important event because the risk of breaking something is greater than the benefit of applying the patch immediately. A vendor may need remote access to troubleshoot equipment. Temporary staff need accounts. A production team may connect equipment that was used at another venue the day before. None of those decisions are unusual.

Problems develop when the temporary decisions are forgotten after the production is over. An account created for a contractor stays active. Vendor access remains open because somebody may need it again. A firewall rule

added during troubleshooting never gets removed. Equipment that was supposed to be temporary becomes part of the normal environment because it worked and nobody had a reason to take it out.

For a major production, I would want engineering and security talking before the event rather than meeting for the first time when something goes wrong. That conversation does not need to become a formal security review of everything involved. It should cover the things that could create a real problem during the production. Who has administrative access? Which vendors can connect remotely? What systems would be difficult to replace? How will people communicate if the normal tools are unavailable?

It also helps to know who gets to make the difficult calls. If suspicious activity appears on a system supporting a live broadcast, somebody may have to decide whether to disconnect it. Security may understand the threat while engineering understands what taking the system down will do to the production. Neither perspective can simply override the other. The decision has to account for both.

Traditional change control can struggle in this environment. Production teams sometimes need to make changes immediately, and a process designed around days of review will eventually be bypassed. I would rather have a simple way to document an emergency change than pretend emergency changes are not going to happen. The important part is knowing what changed and going back afterward to determine whether the temporary change should remain.

Security monitoring has a similar problem. A major event can produce activity that would look unusual on any other day. There may be more people connecting remotely, much larger content transfers, temporary systems, new locations, and vendors working at unusual hours. A security team that knows nothing about the production may spend time investigating perfectly normal activity while overlooking something more important.

That does not mean lowering security monitoring during an event. It means giving the people doing the monitoring enough context to understand what they are seeing. Even a basic production schedule and a list of expected vendors, locations, and significant changes can make alerts much easier to interpret.

Network design can help keep a relatively small problem from becoming a production-wide problem. A temporary laptop used by a contractor does not need the same access as a control system. Vendor connections do not need to provide a general path into the production network. Equipment brought into a venue can be placed where it has access to what it needs without automatically gaining access to everything else.

The design has to be practical. If segmentation prevents the production team from doing its job, people will find ways around it. Broadcast engineers are very good at making things work when the clock is running. Security is better served by giving them a workable path than by creating controls they will have to bypass to get the show on air.

Communication gets difficult once an incident begins. The engineers trying to keep the production running may suddenly have security, management, legal, communications, and other teams asking them for updates. Those requests are reasonable, but answering the same questions for five different groups takes people away from the work that needs their attention.

Having one person coordinate the incident can make a noticeable difference. That person does not need to be the most technical person in the room. Their job is to keep information moving, bring the right people into decisions, and give the technical team enough space to work. During a live event, that coordination can be just as important as another technical tool.

Backup plans also deserve some skepticism. A backup system that nobody on the current crew has used in a year may work perfectly and still be difficult to put into service under pressure. A written manual procedure is

not much help if the instructions or passwords needed to perform it are stored on the system that just became unavailable.

The only reliable way to find those problems is to use the fallback occasionally. That does not always require a large exercise. Have the people who would actually be working the event switch to the alternate process and see what happens. Leave it running long enough to discover whether it can support real production rather than simply proving that it can be turned on.

Live events also bring physical issues back into the security discussion. Equipment is moving through temporary venues. Contractors and crews are working in shared spaces. Network connections may be running through locations the broadcaster does not control. Someone unplugging the wrong cable can have just as much operational impact as a cyber event, and in the first few minutes the symptoms may look very similar.

There will always be more operational flexibility during live production than a security team might prefer. That is part of broadcasting. The useful question is whether the people running the production understand where they have taken additional risk and whether they have a reasonable way to deal with a problem while the clock is still running.

Newsrooms and Editorial Operations Need Their Own Security Model

Newsrooms create an interesting problem for security because journalists are expected to do many of the things employees elsewhere in the company are routinely told not to do. They communicate with people they do not know. They receive unexpected files and links. They spend time on social media, work from hotels and public locations, and connect with sources using whatever communication method the source is comfortable using. That is simply part of reporting.

A security program that does not understand that work can become a problem of its own. Telling a reporter never to open an attachment from someone they do not know is not particularly useful when receiving information from people they do not know is part of the job. Blocking communication platforms may look reasonable from a security perspective until one of those platforms is where sources are contacting the newsroom.

I would design around that reality rather than trying to change it. If reporters regularly receive files that could be risky, give them a safer place to open those files. If someone is working on a story involving a hostile organization, there may be a reason to use a separate device or research environment. If a source needs a more secure way to communicate, reporters should have an option available without assuming every source will know how or be willing to use it.

The threats are also different for reporters and people who regularly appear on air. Their names, photographs, email addresses, social media accounts, and other information are often public because of their jobs. Someone trying to compromise the broadcaster does not necessarily have to begin by attacking corporate infrastructure. Going after a reporter's personal email or social media account may be easier.

That is why generic security awareness training only goes so far for these employees. A reporter is probably better served by knowing how an attacker might impersonate a source, take over a social account, abuse an account recovery process, or use personal information against them. Someone covering a controversial story may also have to think about harassment and threats that would be unusual for most employees.

There is sensitive information inside the newsroom that deserves attention as well. An unpublished investigation can contain source information, interview notes, legal discussions, drafts, and material that

someone may have a strong interest in seeing before publication. Even a production schedule can reveal where people will be and what the organization is planning to cover.

The challenge is protecting that information without making reporting unnecessarily difficult. Journalists need to collaborate, and stories move between reporters, editors, producers, legal teams, and others as they develop. The answer cannot be to lock everything down so tightly that people start creating their own ways to share information. It is more useful to identify the material that really does require additional care and make sure the tools available to the newsroom can support it.

AI is already becoming part of this work. A reporter may use it to transcribe an interview, summarize a document, help translate material, or work through a large amount of research. Those can be perfectly reasonable uses. The concern is what gets sent to the AI service along the way.

An unpublished interview with a confidential source is very different from a publicly available transcript. The same is true of documents connected to an investigation or material that is under legal review. Reporters need enough guidance to recognize those differences without having to consult a policy every time they want to use an AI feature.

The output itself still needs ordinary editorial judgment. A well-written summary can leave something important out. A translation can subtly change meaning. An AI research tool can provide information that sounds credible and is wrong. Those problems become harder to notice precisely because the output often looks finished.

Synthetic media will put even more pressure on the newsroom. During a breaking story, a video or recording may begin circulating before anyone knows where it came from. Security may be able to help examine the file or understand how it was distributed, but the decision to use it remains an editorial one. Those teams will work better together if they already understand their respective roles before that situation occurs.

The reporting itself can also change the threat around the newsroom. Coverage of an election, war, criminal investigation, controversial company, or public figure can attract attention from people who would otherwise have little interest in the broadcaster's systems. Reporters may suddenly receive more phishing messages or attempts to impersonate sources. Social accounts may be targeted. Employees may be harassed or have personal information published.

Security should be able to respond when that happens without becoming involved in deciding what the newsroom covers. If a reporting team is moving into a situation that creates additional risk, security can help protect accounts, watch for impersonation, provide safer equipment or communications, and make sure reporters know who to contact when something unusual happens.

An incident involving a newsroom may also look very different from an ordinary corporate breach. An attacker who gains access to a reporter's account may not care about disrupting operations at all. They may be looking for the identity of a source or trying to learn what the newsroom knows about a story that has not been published.

That changes the way the organization may have to respond. Resetting an account is straightforward. Determining whether a confidential source may have been exposed is not. Legal, editorial, security, and executive leadership may all become involved, and some of the normal assumptions about communicating a security incident may need to account for the broadcaster's responsibilities to its sources.

Newsrooms are always going to be more open to the outside world than most parts of a company. They have to be. Security works better when it starts from that fact and helps reporters handle the risks that come with the job instead of expecting them to behave like employees whose work happens entirely inside the company.

Where Physical and Cyber Risk Meet

Broadcasting has never been a purely digital business. There are studios, control rooms, transmitter sites, cameras, production equipment, network closets, generators, and a lot of other physical infrastructure that has to work for the operation to stay on the air. What has changed is how much of that physical environment can now be controlled or reached through a network.

That creates some fairly ordinary problems that are easy to overlook. Someone who gets into an equipment room may have access to a network port, a console, or a workstation that would be well protected from someone trying to reach it from the internet. A contractor working in a studio may be standing next to equipment connected to systems they would never be allowed to access remotely. An unattended laptop may already have an authenticated session open. Physical access changes the situation very quickly.

This gets harder during live events and temporary productions. There may be people everywhere, and many of them are supposed to be there. Crews are moving equipment, vendors are troubleshooting problems, contractors are running cables, and people are coming and going from spaces that would normally be much more controlled. It can be difficult to tell the difference between someone doing their job and someone being somewhere they should not be.

At the same time, many of the systems used to protect the facility are now technology systems themselves. Door access may depend on a network connection. Cameras may be IP-based and remotely managed. Building controls may be connected to the same broader infrastructure used to monitor other systems. Even something as basic as getting into a secured room can become more complicated if the badge system or the identity service behind it is unavailable.

That is where continuity plans sometimes have an obvious hole. They explain how to recover technology without considering what happens to the facility while that technology is unavailable. If badges stop working, can the right people still get into the control room or equipment space? If the normal communications platform is down, how does the facility team reach engineering or security? If the camera system disappears from the network, does anyone know whether the cameras stopped working or the network simply stopped carrying the video?

Transmitter sites make this even more practical. Many of them are remote and unattended for long periods of time. The ability to manage equipment remotely is necessary because sending someone to the site for every issue would make no sense. When remote access stops working, though, there may be very little information available about what is actually happening.

At that point, someone may have to go to the site. What looked like a technical problem from a monitoring console could turn out to be a power failure, damaged equipment, a telecommunications problem, or evidence that someone has physically been there. The people responding should at least be prepared for that possibility rather than assuming they are being sent to reboot a failed device.

Portable equipment deserves the same kind of thinking. Broadcast equipment moves around constantly. Laptops, cameras, storage devices, encoders, networking gear, and other equipment leave a controlled environment, spend time at a venue or in a vehicle, and eventually come back.

The replacement cost of the equipment is only part of the concern. A laptop may contain credentials or unpublished material. A storage device may hold content that has not aired. A piece of network equipment may still contain configuration information. Other devices may have credentials that allow them to reconnect automatically when they return. Losing one of those devices can create a security problem that continues after the hardware itself is gone.

There is also value in comparing physical and technical information when something unusual happens. If a privileged account is used at an unexpected time, a badge record or camera image may help explain it. If equipment at a remote site suddenly goes offline, knowing that a door was opened shortly beforehand changes the situation. Neither event necessarily means much by itself.

That information is often owned by different teams, which is fine until nobody realizes the other information exists. The answer does not have to be a new platform that combines every physical and cyber alert. In many organizations, it may be enough for the teams to know when to call each other and what kinds of events are worth comparing.

I would keep the responsibilities separate where that makes sense. Physical security, broadcast engineering, facilities, and cybersecurity all have different jobs and different expertise. The issue is what happens in the gaps between them.

A failed badge reader may be a facilities problem right up until nobody can get into a room containing equipment needed to keep the station running. A stolen laptop may begin as a physical security issue and become an identity problem once someone realizes what was stored on it. A network outage at a transmitter site may turn out to have nothing to do with the network. During an actual disruption, the broadcaster has to deal with what happened, regardless of which department would normally own it.

Incident Response Must Be Built Around Decisions

Incident response plans usually look pretty good until there is an incident. They have severity levels, call trees, roles, notification procedures, and pages describing how an investigation should work. All of that is useful. The trouble usually starts when the security team is still trying to figure out what happened and the business wants to know what it should do right now.

Broadcasting makes that problem harder because the operation does not stop while the investigation takes place. The morning show is still coming up. A game may already be on the air. Reporters are working stories and digital teams are publishing content. Advertising is running. People are trying to do their jobs while security is telling them that something they depend on may be compromised.

The first few hours are rarely as clear as an incident plan makes them look. Maybe an administrator account was compromised. That does not immediately tell you what the person behind it did. A vendor might call and say they have a security incident but cannot yet tell you whether your systems are involved. A server may be doing something unusual and nobody knows why. There is usually a period where everyone knows there is a problem and very little else is certain. That is often when some of the biggest decisions have to be made.

Suppose the security team sees suspicious activity involving a system being used for production. They can recommend taking it offline, but that recommendation has a consequence. Production may lose something it needs to stay on the air. The alternative is to leave the system running while the investigation continues. Neither option is particularly comfortable, and waiting may be a decision of its own.

I would want people to know who can make that call before they are standing in a control room arguing about it. Security can explain what it sees and what it is worried about. Engineering can explain what happens if the system is disconnected. Production knows what alternatives are actually available. Someone still has to decide.

That is the part of incident command that matters to me. The incident commander does not need to become the person making every decision in the company. They need to keep the response from turning into five different groups working the same incident independently. They need to know which decisions are waiting, who needs to make them, and what information those people need.

Without that coordination, the technical team can quickly become its own help desk for the incident. An executive calls for an update. Legal wants another call. Communications needs to know what it can say. A vendor wants logs. The insurance carrier needs information. Meanwhile, the people who are supposed to determine what happened are spending much of their time explaining what they do not know yet.

I have always found it more useful to keep the incident picture fairly simple. What do we know right now? What are we still unsure about? What is actually having trouble? What are we worried could be affected next? Is there a decision someone needs to make in the next fifteen minutes? That is enough to keep most people oriented without dragging them through the technical investigation.

The incident can also look very different depending on where you sit. Security may be looking at a compromised identity and consider the technical activity fairly limited. Production may see the same incident as serious because that identity is the only way into a service needed for the next broadcast. A technically complicated intrusion into a system nobody needs today may actually create less immediate pressure.

This is why I would not let a severity number drive the response by itself. The condition of the business matters. If the broadcaster can still operate safely while the investigation continues, there may be time to be deliberate. If people have lost access to something needed for the next hour of programming, the pace changes immediately.

Recovery introduces its own tension. Operations wants the system back. The investigative team may want it left alone. Legal or the forensic firm may want information collected before anything changes. Everyone can be right from their own perspective.

Sometimes the answer will be to capture what can be captured and move on because the broadcast matters more. In another situation, rebuilding too quickly may destroy the only useful evidence of what the attacker did. I do not think a playbook can make that judgment in advance. It can make sure the people making it understand the tradeoff.

Outside responders can complicate this if the organization has never actually worked with them. During a serious event, there may suddenly be outside counsel, forensic investigators, an insurance carrier, technology vendors, a managed security provider, and perhaps law enforcement involved. Each one may need something from the broadcaster.

That is worth sorting out before an incident. Who calls the insurer? Who engages the forensic firm? Does outside counsel need to be involved first? Who can authorize a vendor to make changes? If a cloud provider needs to preserve information, who knows how to request it? These are mundane details until everyone is trying to figure them out at two in the morning.

The same discipline is needed when information moves up to executives. Early findings change. That should be expected. An analyst may say that an account appears to have accessed a system and discover an hour later that the activity came from somewhere else. The mistake is not changing the assessment. The mistake is allowing the first assessment to travel through the organization as though it had already been proven.

I would rather tell leadership that we do not know yet. We can explain what we believe, why we believe it, and what we are doing to confirm it. That gives people something they can use without creating a false sense of certainty.

The most useful review after an incident is often not a discussion of how the attacker got in. That obviously matters, but I would also want to know where the organization struggled once the incident was underway. Did we spend forty minutes trying to find someone who could approve shutting down a connection? Did the

backup process depend on an account that was also unavailable? Did a vendor have more access than anyone realized? Did the technical team lose an hour because everyone wanted their own briefing?

Those are the things I would fix before the next incident. The next attacker may use a completely different technique, and the next outage may not involve an attacker at all. The broadcaster will still have to make decisions with incomplete information while people are trying to keep the operation running. That is the part of incident response a plan has to make easier.

Business Continuity Must Reflect How Broadcasting Actually Works

Business continuity and cybersecurity are still treated as separate programs in a lot of organizations. That may have made more sense when continuity planning was largely concerned with losing a building, power, or some other physical resource. It makes less sense when so much of the business depends on technology that can disappear during a cyber incident.

A ransomware event can make a facility perfectly usable while leaving the people inside it unable to work. An identity problem can lock employees out of applications that are otherwise running normally. A cloud provider can have an outage. A vendor can go offline. Email and collaboration tools can disappear at the same time people need them most. From the perspective of the person trying to get a newscast on the air, it does not really matter which plan owns the problem.

I think this is where some continuity plans become too dependent on the assumption that IT will eventually fix things. The plan describes how the business will respond while technology is being restored, but it may not say much about what happens if restoration takes longer than expected. A cyber incident can make that even more difficult because a system may be technically available and still not be something the organization is comfortable using.

A broadcaster should have some idea of what it can do without its normal environment. That does not mean recreating the entire operation somewhere else. It means understanding what absolutely has to continue and how much of the normal process can be stripped away for a period of time.

A television station, for example, may be able to stay on the air without all of its normal graphics, automation, or production capabilities. A newsroom might still be able to gather and publish important information using a much smaller group of systems. A digital team may be able to put up a basic page with essential information even if the normal publishing environment is unavailable. Those are very different operating conditions, but they may be good enough for a few hours or even longer. The point is to work them out before people are trying to invent the process during an outage.

The details are usually where the fallback begins to fall apart. Someone needs a password that is stored in the unavailable password manager. The backup process requires the same identity provider that everyone is locked out of. The alternate communication method turns out to be another application using the same cloud service. The person who knows how to operate the older system is on vacation. None of those are particularly exotic failures. They are the kinds of things that show up when a team actually tries to use its continuity plan instead of reading through it in a conference room.

People are one of the easiest dependencies to underestimate. Broadcast operations often have individuals who know a particular system extremely well because they have worked with it for years. That is valuable until that person is unavailable and nobody else knows the workaround they normally use when something fails.

Some of that can be addressed through documentation and cross-training, although neither needs to become a major program. If a process is important enough that the broadcaster cannot operate without one particular

person, someone should at least recognize that dependency. The same thinking applies to vendors that hold knowledge the internal team does not have.

Remote work has helped with continuity in some situations, but it has created its own assumptions. Working from somewhere else only helps if people can connect. They still need internet access, credentials, applications, and a way to communicate with everyone else. If the disruption involves the identity platform or the collaboration service everyone uses, sending people home with laptops may not solve very much.

Physical facilities have not disappeared from the equation either. A broadcaster may use a great deal of cloud technology and still depend on studios, control rooms, transmitter sites, production equipment, power, and specialized connectivity. Moving people to another location may preserve some activities while leaving others behind.

I would want to know what actually moves with the people and what does not. A newsroom may be able to work from another location fairly easily. A production that depends on equipment installed in a control room is a different problem. Continuity planning becomes much more useful when those differences are acknowledged instead of assuming that an alternate location solves everything.

Communications is another area I would test on its own. During a significant cyber incident, the normal corporate communication tools may be unavailable or they may be part of what is being investigated. If people suspect that accounts have been compromised, they may not even want to use the normal platform.

The people coordinating the response still need to find each other. Current contact information needs to exist somewhere other than the system that may be down. There should also be a reasonable way for the incident team and executives to communicate if normal accounts cannot be trusted. This does not need to be complicated, but it needs to be something people know how to use.

I would also be careful about declaring a fallback successful because somebody tested it for fifteen minutes. Many workarounds are fine for a short period. The real problems appear several hours later.

Manual work starts to pile up. Information that would normally move automatically has to be entered by hand. People get tired. Someone misses an update. A vendor cannot support the temporary arrangement indefinitely. The workaround that looked reasonable for an hour may become its own source of problems by the end of the day.

That is why duration matters when continuity is tested. If leadership believes a particular process can keep the broadcaster operating for twenty-four hours, the organization should have some evidence that it actually can. Even a partial exercise can expose assumptions that would never appear in a tabletop discussion.

Getting back to normal deserves attention too. During an outage, people may have been working manually, keeping local files, using alternate systems, or entering information somewhere that does not automatically synchronize with the primary environment. Turning the main system back on does not make all of that disappear.

Someone has to determine what changed while the system was unavailable and what needs to be brought back. In some cases, the organization may have two versions of the same information and need to decide which one is correct. Moving back too quickly can create a new problem immediately after the first one has been resolved.

I would judge a continuity program by what the broadcaster can actually keep doing when the things it normally relies on are no longer available. A well-written plan may help, but it does not tell you whether the

workaround really works, whether people know how to use it, or whether it can last as long as leadership assumes. You find that out by trying it before you need it.

Recovery Has to Restore the Business

Recovery discussions can become very technical very quickly. We talk about backup schedules, recovery time objectives, recovery point objectives, replication, and storage. Those are all important, but none of them tell me whether the broadcaster can actually get back to work.

A restored server is not the same thing as a restored operation. The application may be running, but employees cannot authenticate. The data may have been restored, but an integration the application depends on is still unavailable. The production system may be back online, but nobody is comfortable reconnecting it because the team still does not know how the attacker got in. From an infrastructure perspective, something has been recovered. From the broadcaster's perspective, it may still be unusable.

Cyber incidents make this especially difficult because the organization may be rebuilding into an environment it no longer completely trusts. If an administrator account was compromised, simply restoring the affected systems does not address the account the attacker used. If ransomware moved through the network, reconnecting restored systems too quickly could expose them again. Some systems may need to stay isolated while the investigation continues.

That can put recovery teams in an uncomfortable position. The business wants things back as quickly as possible, and that pressure is understandable. At the same time, bringing a system back before the organization understands enough about the incident can create another outage or make the investigation harder.

I would start recovery planning with the things the broadcaster needs to do rather than the list of systems that need to be restored. If the priority is getting a channel back to an acceptable level of service, what does that actually require? Which production systems have to work? What content and configuration data are needed? Which people need access? Does a vendor have to be available? What network and identity services have to exist before any of those things are useful?

That exercise usually exposes dependencies that are easy to miss in a traditional recovery plan. Restoring the production application first may accomplish very little if identity is still unavailable. Restoring a large media archive may be less urgent than recovering a much smaller set of current production data. A system that appears secondary on an infrastructure list may turn out to be required by several of the applications everyone considers critical.

Ransomware adds another issue because the broadcaster has to be reasonably confident that it is not restoring the attacker along with the system. That may mean resetting credentials, rebuilding systems from known sources, checking configurations, applying updates, and watching the restored environment more closely for a period of time.

All of that slows recovery. It should. There is a difference between restoring something quickly and restoring something you are prepared to trust. Continuity plans need to recognize that difference instead of assuming the recovery time measured during an ordinary equipment failure will also apply after a serious security incident. Backups themselves deserve some skepticism. I have seen organizations put a great deal of confidence in the fact that backups are running successfully. The harder question is what happens to those backups if someone gains administrative control of the environment.

If the same credentials can administer production systems and the backup environment, an attacker may be able to reach both. That is one reason organizations use approaches such as immutable storage, separate

administrative accounts, or copies that are not continuously accessible from production. The right approach will vary depending on the system, but there should be some thought given to what the backup is protected from, not just whether it exists. The best way to understand recovery is to actually recover something. Restoring a file proves that a file can be restored. It does not prove that a production application can be rebuilt and used.

A more useful test goes farther. Bring the application back. Load the data. Connect the things it depends on. Have users log in and perform the work they would normally perform. If a vendor has to participate, include the vendor. Then see how long the process actually takes.

That is usually where the interesting problems appear. A license key is missing. The documentation refers to an old version of the application. The person who knows a particular configuration step is not available. An administrator cannot retrieve a password because the password manager is part of the outage. The vendor contact listed in the plan left the company two years ago. Those are recovery problems even though none of them involve backup technology.

Cloud and software-as-a-service applications can create another assumption that needs to be tested. A provider may say that it backs up the service, but that does not necessarily tell the broadcaster what happens during a major outage or how quickly its particular data and functionality will return. For an important service, I would want to understand what the provider is actually promising and what the broadcaster can do while waiting.

Sometimes keeping an independent copy of important data makes sense. In other cases, the better answer may be a temporary process that does not depend on the application at all. There is no reason to build an elaborate technical recovery capability for every SaaS product, but there should be an answer for the ones the business cannot operate without.

This is also why I would be careful with recovery metrics that look impressive on a dashboard. Saying that 98 percent of systems are backed up tells me very little about what happens during a serious incident. I would rather know the last time an important broadcast process was actually recovered and whether it worked.

If leadership believes a critical service can be restored in four hours, there should be some experience behind that number. Maybe the last exercise took six hours because a vendor was slow to respond. Maybe identity recovery added another two hours. Maybe the technical restoration met the target but users could not actually work until much later. Those details tell leadership far more about resilience than the percentage of successful backup jobs.

Recovery is where many assumptions about resilience finally get tested. Architecture diagrams can show redundancy and dashboards can show healthy backups, but eventually someone has to rebuild the environment and use it. Until that happens, the organization has a recovery design. Once it has been tested under realistic conditions, it has some evidence that the design will actually work.

Security Operations Need Broadcast Context

A security operations center can see a great deal of activity without necessarily knowing what any of it means to a broadcaster. An administrator logging into a production system at two in the morning might look suspicious until someone explains that there is a live event underway. A large transfer of media could look like data leaving the company when it is actually part of the normal production process. A vendor connecting remotely might be expected for one event and a serious concern the following week.

That is one of the difficulties with applying a traditional security monitoring model to broadcasting. Security tools are generally good at showing that something happened. They are much less capable of understanding why it happened or whether it makes sense in the context of what the broadcaster is doing at that moment.

The security team needs some familiarity with the operation for that reason. They do not need to understand how to produce a newscast or troubleshoot every piece of equipment in a control room. They should know which systems matter, who normally has administrative access to them, which vendors connect remotely, and when unusual activity might actually be expected.

Broadcast engineering needs some understanding of the security side as well. An engineer may notice something that would never generate a meaningful security alert. A system behaves differently after a vendor connects. A configuration changes unexpectedly. An account appears that nobody remembers creating. A device starts communicating somewhere it normally does not. Those observations may provide the first indication that something is wrong.

This relationship becomes more important because security will never have perfect visibility across a broadcast environment. Some production equipment was not built with modern monitoring in mind. It may provide limited logs or no practical way to install endpoint security software. Other systems may be too sensitive to production changes to treat them like ordinary corporate computers.

I would rather have those limitations clearly understood than pretend the SOC can see everything. If a production device cannot be monitored directly, there may be other places to watch what is happening around it. The network may provide useful information. Administrative access can be routed through a controlled system. Connections into the production environment can be logged. Segmentation can make unexpected communication easier to identify.

The important part is knowing where the blind spots are. A visibility gap that everyone understands can usually be managed. A visibility gap that nobody knows exists tends to become important at exactly the wrong time.

It also changes what the security team should be looking for. A broadcaster can generate enormous amounts of normal activity, and collecting more alerts does not necessarily make the operation safer. I would pay particular attention to changes that affect who controls the environment.

A new administrator appearing unexpectedly would get my attention. So would changes to privileged access, logging being disabled, unusual remote connections, unexpected use of service credentials, or changes to an automation platform that nobody can explain. The same applies to an identity suddenly accessing systems that do not fit the person's normal role. Those events may turn out to be legitimate, but they are worth understanding because they can change what someone is able to do.

Threat intelligence has a place here, although it needs to lead somewhere useful. If a vulnerability is being actively exploited in equipment the broadcaster actually uses, that information may justify changing a maintenance schedule or putting a temporary control around the system. If media companies are being targeted ahead of a major event, the security team may want to pay closer attention to accounts and systems associated with that production.

There is less value in feeding the operations team a constant stream of threat reports that have little connection to its environment. Someone has to make the translation from what is happening outside the company to what, if anything, should change inside it.

Automated response creates a similar issue. Security platforms can disable accounts, isolate devices, block connections, and take other actions very quickly. That can be extremely useful. It can also cause a production problem if the platform has no idea what the account or device is doing at the time.

I would be very careful about automatically isolating something involved in a live broadcast without understanding the consequence. That does not mean security should wait while an attacker moves through the environment. It means the response should be worked out with engineering ahead of time. There may be systems where immediate isolation is appropriate and others where the first action should be to get an engineer and security analyst looking at the same information.

The same problem exists when monitoring is outsourced. A managed security provider may have better tooling and twenty-four-hour coverage, but it does not automatically understand the broadcaster. From the provider's screen, a production server may be little more than a hostname and an IP address.

The broadcaster has to provide some of the missing context. The provider should know which systems would create a serious operational problem if they were isolated, who to contact when something important happens, and which vendors are expected to connect remotely. If there is a major event or unusual production schedule coming up, that may be useful information as well.

Security telemetry can also become useful in situations that do not initially look like traditional security incidents. Suppose questionable content appears through an authorized publishing account. The immediate question may be whether someone deliberately published it, an account was compromised, a vendor integration was involved, or something happened inside an automated workflow.

Authentication records, cloud logs, administrative changes, application history, and file activity may help answer that question. The same can be true when AI is involved in a content workflow. Security operations may not decide whether the content was appropriate, but they may be able to show which account or service touched it and what happened before it was published.

I would measure the security operation with that kind of usefulness in mind. The number of alerts processed during a month tells very little about whether the broadcaster is better protected. A team can close thousands of alerts and still miss the one change that gives someone control of an important production system.

What matters more is whether the team notices significant changes soon enough to do something about them and whether the people responding understand what those changes mean to the operation. During an incident, security should be able to provide information that helps someone make a decision, rather than simply adding another stream of technical data to an already confusing situation.

That is the role I would want security operations to play in a broadcast environment. The SOC does not need to know everything happening in production, and engineering does not need to become an extension of the SOC. They need enough understanding of each other's world that when something unusual happens, they can figure out whether it matters before the broadcaster finds out the hard way.

Governance Has to Reach the Decisions That Create Risk

Most risk in a broadcast organization is not created by a security team. It is created through ordinary decisions about how the business will operate. A production group chooses a platform because it makes the workflow easier. Engineering gives a vendor remote access because the vendor supports equipment the station depends on. Several services move to the same cloud provider because the relationship is working well. A temporary connection stays in place after an event because it may be needed again. An AI tool starts with a small use case and gradually gets connected to more information and more systems.

None of those decisions is necessarily a problem. In many cases, they are exactly what the business should be doing. The risk develops when the organization does not notice how much has changed around the original decision.

A vendor that once supported one application may eventually have access to several production systems. A cloud provider that started with one workload may now support enough of the operation that an outage would affect several departments at once. A workaround created for one live event may still be there six months later. By then, the original decision has become part of how the broadcaster operates, even though nobody made a deliberate decision to accept the larger dependency. Governance should help catch those moments.

That does not mean sending every technology purchase or access request to an executive committee. Most decisions should stay with the people closest to the work. The question is whether the organization has a reasonable way to recognize when a decision has become bigger than the team making it.

Consider vendor access. An engineer allowing a support technician into a system for an hour is routine. The situation changes when the vendor needs permanent administrative access because that is easier for ongoing support. It changes again if the same credentials or connection can reach several important systems. At some point, what began as a support decision has become a business dependency.

The same thing happens with consolidation. Using fewer providers can reduce cost and make an environment easier to manage. It can also mean that one provider failure now affects production, communications, digital publishing, and other services at the same time. Neither outcome makes consolidation inherently good or bad. Someone simply needs to understand how much of the operation now depends on the same place.

This is where broad statements about risk appetite usually fall short. Saying that the organization has a low tolerance for cyber risk does not help anyone decide whether another production service should be placed with a provider the broadcaster already depends on.

A more useful discussion is about the consequence the business is prepared to accept. There may be broadcast functions where losing one provider for several hours is manageable. There may be others where that would create an unacceptable interruption. Some systems may need an independent recovery path because of what they control. Others may have a perfectly reasonable manual workaround.

Those choices should influence where the broadcaster spends money as well. Redundancy is expensive, and building it everywhere would make little sense. A system that can be worked around for a day may not justify another platform or provider. A dependency that can take several important services down at once deserves a different conversation.

The board should see risk at roughly that level. A report showing thousands of vulnerabilities or hundreds of security alerts may say more about the amount of technology in the company than it does about the broadcaster's ability to withstand a disruption.

It is more useful to know that several important services depend on the same identity provider and that the recovery plan for that provider has never been tested. The board should know when a critical vendor has become difficult to replace, when a recovery exercise exposes a serious weakness, or when management has knowingly accepted a risk because fixing it immediately would create a larger operational problem. Those are conditions of the business that directors can understand and challenge.

AI will create many of the same issues. A separate AI committee may be useful while the broadcaster is working through early questions about the technology, but eventually AI becomes part of ordinary operations. At that point, the important questions are not unique to AI.

What information can the service use? What can it access? Can it take action or only make a recommendation? Who approved that access? What happens if the provider changes the service? Who is responsible when the technology is used in an editorial or production process? Those questions already belong somewhere in the organization.

Exceptions need similar attention because broadcasting cannot operate as though every security requirement can be followed all the time. There will be old equipment that cannot support a current control. There will be production deadlines that make a change impractical. There will be situations where a vendor needs access that would not normally be allowed.

The concern is not that an exception exists. The concern is that nobody remembers it exists. If a control has to be bypassed, there should be a reason, somebody responsible for the decision, and some understanding of how long the situation is expected to last. Otherwise a temporary accommodation quietly becomes the normal configuration.

Editorial operations also need a clear boundary. Security may discover that a publishing account has been compromised or that something in a content workflow was altered. It should be able to raise that concern quickly and provide whatever evidence it has. That does not make security the editor.

The newsroom still decides what it publishes. Security's job is to help establish whether the systems, accounts, and information involved can be trusted. Keeping that distinction clear protects both the broadcaster and the independence of the editorial process.

There is a broader accountability issue underneath all of this. Cybersecurity has become involved in so much of the organization that it is easy for every technology-related risk to end up assigned to the security team. That does not make much sense when security did not make the underlying business decision.

If a leader chooses a vendor because the service is important to the business, that leader has a role in accepting the dependency that comes with it. If a production group needs an exception to keep operating, the operational owner should understand what is being accepted. Security can help people see the risk and determine whether there are reasonable ways to reduce it. It cannot absorb responsibility for every decision that happens to involve technology.

This is the part of governance that matters most. Risk is often created a little at a time through decisions that make sense individually. The organization needs to notice when those decisions begin to add up to something more consequential. That is when a routine operational choice becomes something the business should make with its eyes open.

Cyber Insurance Is Part of Financial Resilience

A serious cyber incident creates costs long before anyone knows what the final loss will be. Outside investigators may be needed. Lawyers may become involved. Systems may have to be rebuilt. Vendors may charge for emergency support. Production teams may need alternate equipment or services just to keep operating. Revenue can also be affected while all of this is happening.

Cyber insurance can help with some of that financial exposure. What it covers depends on the policy, and there will always be costs the broadcaster keeps. The important point is that insurance should be considered alongside the other decisions the organization makes about how it expects to get through a serious disruption.

This becomes easier to see when the conversation starts with the loss rather than the policy. Consider a broadcaster that suffers a ransomware incident but manages to stay on the air. At first glance, there may not

appear to be a traditional business interruption because programming never completely stopped. The reality could be very different.

Normal production may have been unavailable for several days. Additional people may have been brought in. Alternate technology may have been rented or purchased. Advertising commitments may have been affected. Digital services may have been unavailable even though the broadcast continued. Legal, forensic, recovery, and communications costs may all be accumulating at the same time. That is the financial side of degraded operations, and it is worth understanding before an incident occurs. Staying on the air does not necessarily mean the broadcaster avoided a significant loss.

The same issue appears with third parties. A broadcaster may be doing everything reasonably expected inside its own environment and still suffer a loss because an important provider goes down. If the identity provider is unavailable, the broadcaster may lose access to applications it needs. If a cloud production service has a serious incident, workflows may have to move somewhere else. A telecommunications failure can create another set of costs even though nobody compromised the broadcaster directly.

The dependency work done elsewhere in the organization should inform the insurance discussion. If a handful of providers can materially affect the operation, finance and risk management should know which ones they are. The question then becomes whether the financial effect of losing one of those providers is something the broadcaster can absorb, something it wants to insure, or something that needs another form of protection.

This is also why the policy should be understood before anyone needs to use it. The incident team does not need to memorize an insurance contract, but somebody should know what has to happen when a potentially covered event occurs.

Who contacts the carrier? How quickly does that need to happen? Does the broadcaster have to use particular forensic firms or legal counsel? Can management engage a provider immediately when production is at risk, or does the insurer need to approve the expense first? Those are much easier questions to answer during an annual review than during the first night of an incident.

The security team has a role here because many of the questions insurers ask are questions security should already be able to answer. Underwriters want to know how administrative access is protected, whether MFA is actually enforced, how endpoints are managed, how backups are protected, and whether the organization has exercised its response and recovery processes.

There is a meaningful difference between answering those questions from policy and answering them from experience. “We perform backups” is not the same as knowing that a critical production service was successfully restored during the last exercise. “We require MFA” is less useful than knowing where it is enforced and where exceptions still exist.

That evidence should already have value inside the broadcaster. It tells management whether the controls it depends on are really working. The fact that the same evidence can support an insurance renewal is useful, but it should not be the reason the evidence exists.

Renewal can still expose problems. It is not unusual for an insurance application to move through several departments because no one person knows all the answers. That is understandable in a complicated environment. It becomes more interesting when different groups give different answers to the same question.

Security may believe a control covers the entire company while engineering knows there are production systems outside it. Finance may believe backups provide a particular recovery capability while the technology team has never tested that scenario. Risk management may list a vendor as replaceable while operations knows

losing that vendor would create an immediate problem. Those disagreements are worth resolving even if they have no effect on the premium. They show where the organization has different versions of its own risk.

The policy also has to match the way the broadcaster could actually lose money. Business interruption is a good example because broadcast operations do not always move cleanly from running to stopped. A station may continue operating at reduced capacity. One distribution channel may fail while another remains available. Additional costs may be incurred specifically to prevent a larger interruption.

Finance should understand how the policy it purchased would treat those circumstances. That conversation belongs with the broker, carrier, and appropriate legal advisers because policy language differs. The objective is simply to avoid discovering after the event that the organization's assumption about coverage was different from what the contract actually provides.

Insurance limits deserve the same practical treatment. Buying more coverage is not automatically the right answer, and neither is minimizing premium. The useful question is how much loss the organization is prepared to carry itself.

A broadcaster that could absorb a $250,000 event without materially affecting the business is in a different position from one that would struggle with the same loss. A disruption involving a major production platform may create a very different exposure from an incident involving a smaller corporate system. Those differences should have some influence on the amount of risk the organization retains and the amount it tries to transfer.

Insurance therefore sits alongside recovery, continuity, and incident response rather than replacing any of them. The broadcaster still has to make the operational decisions. It still has to restore systems, manage vendors, communicate with employees and the public, and work through the disruption.

The policy helps with a different part of the problem. It gives the organization another way to absorb some of the financial consequences when the controls, plans, and recovery capabilities are not enough to prevent a loss. For a broadcaster thinking seriously about business survivability, that financial capacity matters just as much as getting the technology running again.

Measure What Tells You Something

Cybersecurity has never had a shortage of things to measure. Security teams can report how many vulnerabilities are open, how quickly systems are patched, how many alerts were investigated, how employees performed on phishing tests, and how many endpoints have the expected security tools installed. Most of those numbers have some value. The problem is figuring out what they actually tell you about the broadcaster. A dashboard can improve for months without answering a fairly basic question: are we in a better position to handle a serious disruption than we were before?

That question is harder to measure because it cuts across several parts of the organization. A vulnerability scanner cannot tell you that the backup production process depends on the same identity service as the primary process. An endpoint coverage number does not tell you whether the team can operate without a critical vendor. A phishing score says very little about whether an administrator can regain control of an environment after privileged accounts have been compromised. Those are the things I would want measurement to help uncover.

For a critical broadcast service, start with what it depends on. If the organization says the service can be recovered, when was that actually tested? If there is a fallback, has anyone used it recently? If a vendor is necessary to restore the service, does the recovery assumption include the time it takes to get that vendor

involved? These are not complicated measurements, but they say something useful about whether the capability is as resilient as everyone believes.

Time can tell you quite a bit as well. During an exercise, how long did it take before the team understood which part of the operation was affected? How long did it take to get trusted administrative access after the scenario assumed an account compromise? If production had to move to a fallback, how long did that really take?

The answers may not line up with the targets in the plan, and that is useful information. If a recovery objective says four hours and the last exercise took seven, reporting four hours because that remains the approved target does not help anyone. The seven-hour result is the information leadership needs.

I would treat evidence the same way. There is a big difference between having a control and knowing something about how it performs. A policy requiring privileged access reviews tells you what should happen. A completed review tells you that somebody actually looked. Finding an old vendor account during that review tells you even more because now you know the process found something worth fixing.

This is an important part of Integrated Assurance. Confidence should come from what the organization can demonstrate, not simply from the existence of a policy, control, or technology. Sometimes the evidence will show that something works well. Other times it will show that an assumption was wrong. Both results are useful.

That creates an interesting problem with dashboards. Organizations naturally want green indicators, especially when the information is going to executives or the board. Teams quickly learn what produces green. If the measurement system makes a failed recovery exercise look like poor performance, people have an incentive to design easier exercises or avoid testing things they are unsure about.

I would rather see a recovery exercise fail and expose a dependency nobody understood. The failure gives the organization something it can fix. A green status based on a recovery process that has never been seriously tested provides much less comfort.

The same applies to trends. A temporary increase in reported risk may actually mean the organization is getting better at finding problems. More exceptions may appear because someone finally reviewed them. Vendor risk may look worse after the broadcaster maps its dependencies and discovers how much of the operation relies on a small number of providers. Those numbers need explanation. Otherwise, improvement in visibility can look like deterioration in security.

This is especially important when information reaches the board. Directors should not have to interpret a security operations dashboard to understand whether the broadcaster has a problem. They need to know what management is concerned about and whether anything important has changed.

Maybe the last recovery exercise showed that an important workflow takes twice as long to restore as previously believed. Maybe a major vendor has become more difficult to replace. Maybe the organization closed a long-standing gap around privileged access. Those are meaningful changes in the condition of the business. The underlying technical metrics can support the discussion when they are needed.

Operational teams should get something useful from measurement too. An engineer is more likely to care that a change reduced recovery time by two hours than that it improved an enterprise risk score from 3.2 to 3.6. A production team can understand that a fallback process now works without the primary identity provider. Those results connect the security work to something people can see and use.

There will still be a place for vulnerability counts, patching statistics, endpoint coverage, phishing results, and the other measures security teams already collect. Some are needed to manage the program day to day. Others help identify changes that deserve attention. They become less useful when they are presented as evidence of resilience without showing how they affect the operation.

The broadcaster does not need to measure everything it can collect. It needs enough information to challenge its assumptions and decide where something needs to change. If a metric cannot help someone understand a problem, make a decision, or see whether an earlier decision actually worked, it is worth asking why it is on the dashboard at all.

Integrated Assurance as the Operating Model

Integrated Assurance came from a fairly simple observation. Organizations have become very good at dividing risk into areas they know how to manage. Cybersecurity has its responsibilities. Risk has its process. Audit has its role. Engineering, operations, finance, legal, and business continuity all have their own responsibilities as well. There is nothing inherently wrong with that. You need people who understand their part of the business and are accountable for it. Where I think we get into trouble is assuming that because each of those areas is being managed, we understand the risk to the business.

Broadcasting is a good example of why that assumption can be dangerous. Think about something as common as identity. It may be managed by IT and monitored by security, but production systems depend on it. Vendors use it. Employees working remotely depend on it. Cloud services may be tied to it. Recovery may depend on it too. If the identity environment is compromised, the problem does not stay with the team responsible for identity. It moves very quickly into the operation.

The same thing happens with vendors. Procurement may own the commercial relationship, security may have assessed the vendor, and engineering may be responsible for the technology. What is sometimes less clear is how dependent the broadcaster has become on that company. That usually happens over time. The vendor starts with one service, then another workflow gets added, remote support becomes normal, and eventually the relationship is much more important to the operation than it was when the original decision was made.

I have seen the same problem with recovery. An organization can have backups, recovery objectives, continuity plans, and people assigned to all of them and still discover during an exercise that the recovery process depends on something nobody included in the scenario. Maybe it is an identity service. Maybe it is a vendor. Maybe it is a person who knows how to rebuild a particular system. The individual pieces of the recovery program can all exist while the ability to recover is still uncertain. That is really where Integrated Assurance starts. I am interested in what happens when all of these things come together around something the business actually needs to do.

For a broadcaster, that might be producing a newscast tonight. What has to work for that to happen? The answer is probably much larger than the production system itself. People need access to systems and content. The newsroom needs information. Production needs equipment and applications. Some of those applications may depend on cloud services. Vendors may be involved. Communications have to work. If something fails, somebody needs to have the authority to make a different decision.

Once you look at the business that way, you start finding dependencies that do not show up very well when risk is viewed one function at a time. You may find that two supposedly redundant services use the same provider. You may discover that the backup process requires the same credentials you expect to lose during the incident. You may find that a vendor has become far more important than its contract would suggest. None of

this means the organization made a bad decision. It means the business changed and the understanding of the risk did not necessarily change with it.

This is why I do not think assurance should stop at whether a control exists. I want to know what gives us confidence that it will work when we need it. If we say a production service can be recovered in four hours, have we actually done it? If we say a vendor has limited access, when did we last look at that access? If we have a fallback process, has anyone tried to run the operation that way recently?

Sometimes the answer is going to be no, and I am comfortable with that. I would rather know that we are relying on an assumption than report the assumption as though it has been proven. Once the uncertainty is visible, leadership can decide whether it matters enough to test, fix, fund, or simply accept.

That is also why I do not see Integrated Assurance as a cybersecurity program. Security is part of it, but security cannot answer all of these questions. Engineering knows things security does not know. Operations sees what happens after a system is put into use. Risk and audit can challenge assumptions that have been around long enough that people have stopped questioning them. Finance understands consequences and tradeoffs that may not be obvious from a technical discussion. Leadership has to decide which risks are worth accepting because eliminating all of them is neither realistic nor affordable.

Governance matters in that environment because decisions need owners. Architecture matters because decisions eventually become dependencies. Operations matters because it shows whether those decisions are working in practice. Measurement is useful when it tells us something we did not already know. Culture matters because the entire model breaks down if people are reluctant to say that something is not working the way everyone thinks it is. I do not view these as a sequence of activities. They are simply different places where we learn something about the same business.

This is especially important in broadcasting because there will always be exceptions and workarounds. There has to be. Live production does not stop because the normal process is inconvenient. Engineers solve problems. Production teams find another way to get something done. Vendors are brought in because they know the equipment. A temporary connection gets created because there are twenty minutes until air. Most of those decisions are completely reasonable in the moment.

What matters is what happens afterward. Does the temporary access get removed? Does somebody recognize that the workaround has become the normal process? Does the organization learn that it is now dependent on a vendor in a way it was not six months ago? That is where I think a lot of risk quietly accumulates.

Integrated Assurance is meant to make more of that visible while there is still time to do something about it. It does not require every problem to be fixed, and it certainly does not require every decision to go through security. Sometimes the right business decision is to accept the risk because the alternative costs too much or interferes with the operation. I just want that to be an informed decision rather than something the organization discovers during an incident.

For me, that is also a better way to think about maturity. I am less interested in how many security products a broadcaster owns or how many controls it can check off. I want to know how well it understands the things its important operations depend on. I want to know whether its confidence in recovery comes from actually recovering something. I want to know whether information from incidents, exercises, engineers, auditors, and day-to-day operations changes the way decisions get made.

Perfect visibility is unrealistic in a business that changes as quickly as broadcasting. What matters is getting better at recognizing where the organization is relying on something it has not really tested or understood, and making that uncertainty visible before an incident makes the decision for you.

That is how I think about Integrated Assurance. It connects the different views of risk already present in the organization so leadership has a more credible understanding of what the business can actually withstand. For a broadcaster, the test is practical: when something the operation depends on stops working, how much confidence is there that the business can keep doing what it exists to do, and what evidence supports that confidence?

90-Day Agenda for Broadcast Leadership

Improving resilience does not have to begin with a large transformation program. In fact, I would avoid starting there. A broadcaster can learn a great deal in ninety days by taking a few important parts of the operation, understanding what they actually depend on, and testing some of the assumptions that have developed around them. The point is not to produce another cybersecurity roadmap. It is to give leadership better information about where the business could get into trouble and what deserves attention first.

I would start with a handful of things the broadcaster absolutely needs to be able to do. That will be different for every organization. It could be keeping a channel on the air, producing a live newscast, publishing breaking news, supporting emergency communications, maintaining digital distribution, or keeping advertising operations running. There is no reason to begin with twenty processes. Pick the ones where an extended disruption would create a real business problem and learn something from them.

Then sit down with the people who actually know how those activities work. Ask what they need to get the job done and what would happen if some of those things were unavailable. The conversation will move quickly from the obvious production systems into identity, networks, cloud services, data, facilities, vendors, communications, and particular people who know how something works. That is usually where the useful information starts to appear.

A perfect dependency map is not necessary, and trying to build one can consume the entire ninety days. Concentrate on the relationships that could materially change the broadcaster's ability to operate. Several important workflows tied to the same identity provider deserve attention. So does a recovery process that depends on a particular vendor being available, or a fallback that relies on another service likely to be affected by the same incident.

I would spend some time on privileged access while doing this because administrative access tends to cut across many of these dependencies. Look at who can make significant changes to the systems supporting the capabilities being reviewed. Include vendor accounts and service accounts, not just employees with administrator titles. There will probably be accounts that made sense at one time and no longer do. There may also be emergency access that everyone assumes will work but nobody has tried recently. Ninety days is enough time to clean up some of that and test whether the organization can regain trusted administrative access when the normal method is unavailable.

AI belongs in the same review, although I would not try to inventory every employee who has experimented with an AI tool. That would create a lot of work without necessarily telling leadership much. I would look for AI that is already part of a meaningful workflow, has access to company information, connects to another system, or can take an action on someone's behalf. Those are the uses where ownership and permissions start to matter. The exercise will probably uncover more AI use than expected, which is useful by itself.

I would also recover something during the ninety days. Choose one of the important capabilities identified at the beginning and see what it takes to bring it back. Do not stop when a server boots or a backup restores successfully. Get far enough that someone could actually perform the work the business expects that capability

to support. Keep track of what slows the team down because those delays will often tell you more than the recovery target written in the plan.

The same period is a good time to run an incident exercise, particularly after some of the dependency work has been completed. I would build the scenario around what the team has already learned rather than use a generic ransomware tabletop. If the broadcaster depends heavily on one identity platform and a production vendor, put pressure on both. Introduce enough uncertainty that people have to decide whether they trust the environment and whether production should continue. If content integrity is relevant, add something questionable into the workflow and see who is expected to make the call.

The exercise should tell you where the organization hesitates. Maybe nobody is sure who can shut down a vendor connection. Maybe security recommends isolating a system and production cannot explain what happens if it does. Perhaps legal, communications, and the technical team have different understandings of what has actually been confirmed. Those are useful findings because they show where a real incident is likely to lose time.

By this point, the third parties worth examining should be fairly obvious. I would not spend the quarter sending security questionnaires to every supplier. Start with the companies that showed up repeatedly while tracing important broadcast capabilities. Understand what access they have, what the broadcaster expects them to do during an outage, and whether there is any realistic way to operate without them for a period of time. If several critical vendors turn out to depend on the same underlying provider, that is worth understanding too.

The insurance review can be just as practical. Someone on the incident team should know how the cyber carrier is contacted and what happens after that call. They should know whether the policy expects particular legal or forensic providers to be used and where the retention begins. If the dependency work has identified outside providers that could create a significant interruption, finance and risk management can use that information to have a more specific conversation with the broker about how the policy treats that exposure. There is no need to turn this into a separate insurance project.

At the end of the ninety days, I would keep the executive discussion fairly simple. Show leadership the business capabilities that were examined and what the organization learned about them. Explain where an important dependency was found, where recovery was actually demonstrated, and where confidence still rests mostly on an assumption. There will probably be some issues that can be fixed quickly and others that require money, architecture changes, vendor discussions, or an explicit decision to live with the risk.

By then, the broadcaster should know enough to decide what deserves attention next without turning the exercise into a massive project list. The useful result is a better understanding of several important parts of the business and some evidence about whether the assumptions surrounding them hold up. The same approach can then be applied to other capabilities as part of normal planning, investment, and change.

Conclusion: Protecting the Ability to Broadcast

Broadcasting has changed enough that the way we think about cybersecurity has to change with it. A broadcaster is still responsible for producing and distributing content, but the number of things required to make that happen has grown considerably. Identity services, cloud platforms, software providers, remote access, data, outside vendors, automation, AI, communications, and physical infrastructure are now mixed into everyday operations. Most of those changes have made broadcasting more capable. They have also made it harder to separate a technology problem from a business problem.

That is why I think the discussion needs to move beyond protecting systems. Systems obviously need to be protected, but the more important question is what happens to the broadcaster when one of them is no longer

available or can no longer be trusted. Can the newsroom still publish? Can production continue? Can someone establish trusted administrative access? Can the organization determine whether content is legitimate? Can it communicate internally and externally? Can it recover enough of the operation to keep doing business while the larger problem is being worked?

Those questions become more important as AI finds its way into normal work. Much of that use will be helpful and fairly routine. Some of it will eventually have access to systems and information that matter, and agentic AI will be able to do more on behalf of users without someone directing every individual action. Broadcasters will have to understand where that authority exists and what happens when an AI system does something unexpected. The same technology is changing the content environment, where determining what is authentic may depend as much on knowing where something came from and how it was handled as it does on detecting whether AI was involved.

Cloud and third-party services create a similar issue. Broadcasters are not going to stop using them, nor should they. The concern is knowing when a useful service has quietly become something the operation cannot function without. That dependency may be perfectly acceptable, but it should be understood. A vendor outage, cloud failure, identity problem, or communications disruption should not be the first time the organization realizes how much of the business was tied to one provider.

This is also why I have placed so much emphasis on recovery throughout this paper. We can reduce the likelihood and impact of an incident, but we cannot build an environment where nothing fails. At some point, a broadcaster has to be able to work through a situation in which the normal environment is not available. That may mean operating with fewer systems, using a different workflow, restoring services in an unfamiliar order, or making a business decision before the technical team has finished determining exactly what happened. Those are difficult things to improvise during an incident if nobody has worked through them beforehand.

Integrated Assurance fits into this because these problems do not belong to one department. Security has an important role, but so do engineering, production, editorial, finance, legal, risk, audit, business continuity, and executive leadership. Each sees something different, and the broadcaster needs those different views. The problem begins when information stays inside those functions and nobody connects it back to what the business is trying to accomplish. A technically sound decision can still create an operational dependency. A reasonable vendor decision can eventually create concentration risk. A recovery plan can look complete until someone actually tries to use it.

For me, business survivability is where all of this comes together. I use that term deliberately because resilience can sometimes become another program or another score. Survivability is more practical. Something has gone wrong, an important part of the environment is unavailable, and the organization still has a business to run. The question is how much of that business it can continue, for how long, and whether the people making decisions understand what they are dealing with well enough to choose what happens next.

A perfect defense is neither realistic nor particularly useful as an objective. A broadcaster needs to understand the things it depends on and be willing to test the assumptions around them. Leadership should know where recovery has actually been demonstrated, where uncertainty remains, and when a decision involving technology, a vendor, AI, or a production workflow has changed the risk enough to deserve another look.

Broadcasting will keep changing. There will be more software involved, more services outside the broadcaster's direct control, and more automation making decisions or taking actions that people handle today. Cybersecurity has to develop alongside that change without becoming an obstacle to it. The organizations that do this well will not be the ones that managed to put a security control around everything. They will be the

ones that understand their business well enough to know where control matters, where they can tolerate disruption, and what they will do when something they depend on is no longer there.

Cybersecurity has an important role in broadcasting, but protecting technology is only part of the job. The larger responsibility is preserving the broadcaster's ability to operate when the conditions it normally depends on have changed. An organization that understands those dependencies, tests the assumptions behind them, and has practiced how it will respond is better prepared to keep doing what the business exists to do when the normal environment is no longer available.

About the Author

Patrick M. Hayes is an enterprise security strategist, author, and creator of the Integrated Assurance® operating model and Integrated Assurance Maturity Model. His work focuses on connecting cybersecurity, enterprise risk, governance, cyber insurance, and business survivability so organizations can make better decisions about the technologies and dependencies they rely on. He is the author of Integrated Assurance: Unified Risk Strategy, Relevant Impact, and Can We Insure This?