THE BROADCAST BRIDGE
Connecting IT to Broadcast

# Making Cloud Systems Secure

# Essential Guide

EG

ESSENTIAL GUIDES

# Introduction

By Tony Orme, Editor at The Broadcast Bridge

**The conversation around security seems to have come out of nowhere and is gathering momentum. And as more broadcasters take advantage of IP, then the issues regarding security are gaining much more scrutiny.**

IP is an enabling technology, it not only provides a convenient method of media transport, but also facilitates access to many other methods of signal processing and media storage. Without a doubt, IP is delivering unparalleled flexibility and scalability for broadcasters. However, these advantages highlight the challenges around security requiring broadcasters to look at the new methods of securing their on-prem and off-prem datacenters.

Security is a generic term that encapsulates many topics from keeping hostile actors out of an infrastructure, to containing the effects of user error. This might include a perfectly legitimate user deleting a file by mistake, or inadvertently moving files in a playlist without realizing they've done it. Therefore, security is fundamentally about keeping data safe and accessible for only those who have legitimate access to it, as well as being able to recover from innocent errors, or a cyber-attack.

With ever increasingly complex infrastructures and data management systems, the possibility of human error is something that cannot be ignored. Consequently, valuable data must be backed up and be restorable. This leads for the need to rigorously and regularly test processes, and not wait until a file corruption or loss happens for real, only to find there were errors in the process which made the data unrecoverable.

Many of our approaches to security have been based on a highly contained view of how IT systems operate. Traditionally, the areas of access a user has to an IT infrastructure have been greatly limited, usually in the form of physical onsite terminals with logon credentials. But as users demanded more access to the outside world, then wrinkles started to occur in our security thinking, especially when the internet was connected to the infrastructure.

The internet has probably been one of the greatest human achievements in living memory. Not only has it proved to be an amazingly collaborative scientific tool, but it connects people all over the world so they can share ideas, memories, and experiences. It should only be natural then that broadcasting both migrates to the internet as a business tool and uses it as an entertainment transmission method for viewers. Connecting to the internet is a massive win for broadcasters, and a considerable security challenge.

The contained thinking for IT security such as the perimeter wall approach, has served us well for many years as IT engineers have honed their skills in keeping networks and infrastructures secure, and no doubt these methods will continue to serve us well. However, as more users demand greater access to the internet then we must expand the traditional security approach and provide systems that are relevant to today's complex and interconnected networks.

Tony Orme.

Broadcasting is moving to IP, and as we do so, we must take security very seriously. Reassuringly, the tools to achieve exceptionally high levels of security are already out there and so we must look to see how other industries are achieving this, learn from them, and adopt the best of their practices and methodologies.

Tony Orme
Editor, The Broadcast Bridge

# Making Cloud Systems Secure

By Tony Orme, Editor at The Broadcast Bridge

**Security for cloud and internet systems is playing an ever-increasing role in broadcast infrastructures. High value media assets and communication channels to broad audiences are at risk so it is reasonable to assume that unidentified hostile actors are lurking in every corner.**

The good news is that there is much a broadcaster can do to help protect themselves from attack. Although no system can ever be completely secure, it's worth remembering that even traditional SDI and AES broadcast facilities had their vulnerabilities. They were just different, and broadcasters assumed they knew where they were, but they often didn't.

Managing and understanding risk is key for maintaining security. Furthermore, a vast array of detection and analysis tools are available to help broadcasters understand network and infrastructure vulnerabilities, especially as the IT industry has been working at finding solutions to these challenges for many years.

Delivering effective security not only relies on our technical understanding but it also embraces the attitudes of users and how they approach security. To be effective, security must encompass a positive and productive mindset that is promoted and encouraged from the CEO so that it manifests itself as a culture throughout the broadcast facility.

As one of the biggest vulnerabilities to any IT system is human error, effective cloud security is a way of life that must be encouraged. And systems need to be designed to have the right processes, features, and patches in place from the beginning and then throughout the lifetime of the system.

## Problem to Solve

To deliver cloud security, broadcasters need to achieve the following:

1. Protect data storage, processing systems, and networks from data theft.

2. Develop a data recovery plan in case data is lost or corrupted.

3. Stop human negligence so that data cannot be compromised.

4. Ringfence the impact of data loss or a compromise of the system.

Although much of our approach to cloud security revolves around stopping malicious actors from penetrating the network in the first place, we must also be mindful of the need to back up data and be able to recover from data loss or corruption.

Intuitively, we may want to treat data recovery separately from stopping intrusion, but in many instances, there is a great deal of overlap. Furthermore, data loss or corruption may not be a consequence of a malicious act, but instead be a simple mistake such as a user deleting a master media file. The fact that users shouldn't be able to make these mistakes falls into the discipline of user access rights but restoring the media asset should be a key part of the data recovery plan.

Restoring a file is important but it's only a part of the equation. Equally important is isolating a hacker's ability to disrupt an ongoing service. This is where system redundancy and the ability to fail over to a back up system are important.

Cloud systems have the potential to make data recovery much easier due to the multitude of options available for storage. High speed near-line storage can be archived to off-line storage, which is often cheaper but slower. However, these storage systems also need to be protected from malicious attacks. Even if a broadcaster archived their cloud storage to on-prem, the two systems are intrinsically linked, and adequate security must be maintained between the two.

Again, we need to address protecting active processing, not just storage.

## Outdated Approaches

Traditional methods of IT security used the perimeter wall approach. That is, the access points to the network were heavily guarded so that if a hostile actor tried to gain access, they could only do so through a limited number of points that could be protected. One example of this is the firewall on the internet router.

Firewalls and intrusion detection systems would be placed at the internet connection point to the ISP and any malicious access could be detected and stopped. But the fundamental challenge with this strategy is that it relies on knowledge of the attack pattern which can only be gained if another organization had been subject to the attack, detected it, noted the pattern, and then shared it. That said, this is still a very important part of infrastructure protection.

The challenges of the perimeter method have been further compounded in recent years as users become more reliant on the internet. Bring your own devices and the reliance on cloud systems has further exacerbated the inadequacies of this approach. Once the attacker is in the perimeter wall, they can cause all kinds of havoc, sometimes lying-in wait for weeks or even months before releasing their attack.

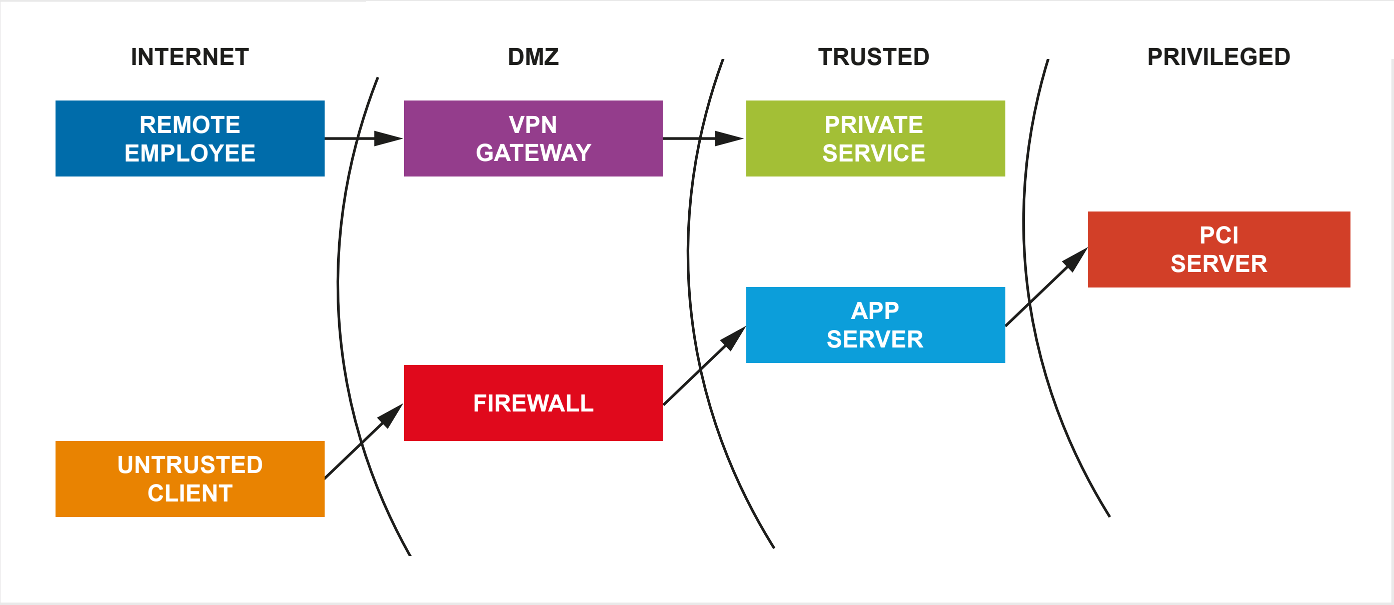


Figure 1 – Traditional perimeter approaches to network and resource security are flawed in modern cloud and IT infrastructures as they lack intra-zone traffic inspection, lack flexibility, and have single points of failure.

As IT and datacenter infrastructures increase in complexity, the need to improve our approach to security has become clear. This isn't just a matter of improving antivirus software or increasing our ability to detect rogue traffic in a network (although these are clearly very important) but is also about adopting a new mindset that makes the assumption that nobody and no-transaction can be trusted.

## Encryption

Stored data is often encrypted so that if an unauthorized user does access the storage, then they will be unable to use the data. They will certainly be able to access it, but they won't be able to decode and view it.

Data is not only vulnerable when it is stored, but also when it is in transit between processes or if the entire platform is under attack. Anybody eavesdropping on the network will be able to gain a whole host of information about the infrastructure leading to a potential attack.

Exchanges between cloud software including microservices often use the RESTful API method. As public and private clouds are connected via the internet then their communication protocols must comply with internet standards.

The REST (Representational State Transfer) protocol provides methods and standards for computer systems on the internet to allow them to exchange data and therefore communicate with each other.  Although this is a massively versatile system, it uses a protocol based on plain text, meaning without additional measures it is highly insecure. Anybody with a network sniffer will be able to view the messages and gain a great deal of knowledge about the sending and receiving networks. And this is especially worrying as the communications are being freely exchanged across the open internet.

This leads to another potential issue, and that is one of man-in-the-middle attacks as any end point using the RESTful API can be impersonated by a malicious actor. This scenario mainly occurs as end point validation was not built into the original web HTTP (Hyper Text Transfer Protocol) specifications. A malicious actor could intercept the traffic on the network and change the IP address of the destination to their own server and then force all the traffic to it, and once they've done that, they can easily harvest the user's credentials.

To alleviate both these challenges, a method of validating the API endpoints was developed using public-private key encryption. This resulted in the adoption of HTTPS (Hyper Text Transfer Protocol Secure) which uses TLS (Transport Layer Security) as its underlying security method. HTTPS solves three challenges: confidentiality, authenticity, and integrity. Confidentiality stops anybody snooping on the connection as it is encrypted so that all sensitive data is obscured. Authenticity guarantees the sender and receiver are who they say they are (thus stopping man-in-the-middle attacks). And Integrity guarantees that the data exchanged between the endpoints hasn't been tampered with or modified.

## OAUTH 2.0

Although HTTPS provides a reliable method of data exchange for APIs, and services such as Active Directory provide authorization, they do not deliver any method of limiting access to cloud services. User credentials achieve access to some extent as the user provides a unique username and password to log onto a service, but to achieve adequate levels of security, they must logon for every service they access. This is often possible for simple website access but is not a reliable method for API connectivity, especially when considering microservices as they can often establish multiple connections at the same time to multiple services.

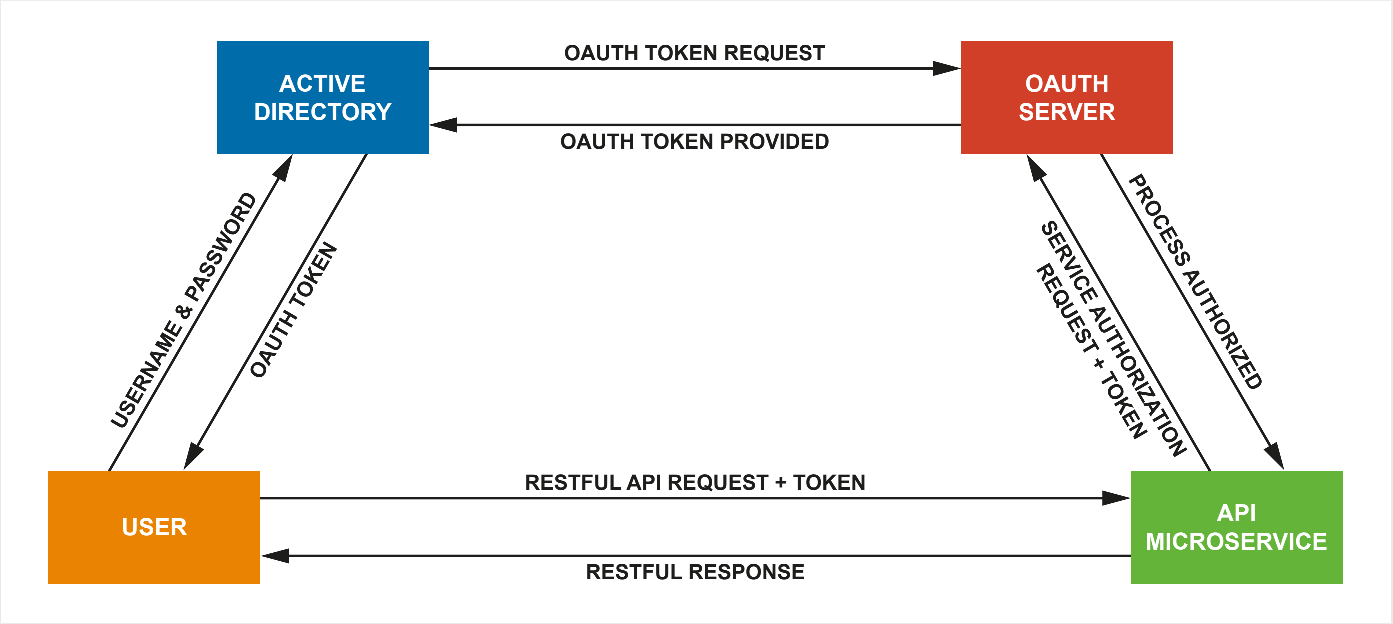


Figure 2 - A session showing how a user is issued with an OAuth token. Initially the client logs onto the network and is validated by the AD, which in turn requests an OAuth token from the OAuth server (assuming it passed AD). When the user requests a service through an API call it issues its token within a JWT (see text), the service then validates the token against the OAuth server and assuming the token is valid, the API is authorized to continue its task.

OAuth uses a method of tokens that both replace the logon credentials of the user as well as providing scope for the service they are connecting to. For example, a transcoding microservice will receive a RESTful API instruction from a user who wants to transcode a file using a transcoder microservice. Included in the API message will be a token that was created by the OAuth authorization server and embedded in this token will be information that defines the resource scope such as file access rights. In this instance, the user may have access to read the input media file but not write to it, thus negating the possibility of the user deleting or corrupting the media file.

Furthermore, if the system administrator wants to stop access to a specific resource, then they can kill the token. Although it may physically exist as a data string, the OAuth authorization server will stop any servers trying to validate it and therefore restrict access to the resource.

The OAuth tokens, as well as embedding resource access information, also have a limited time value. That is, after a predefined length of time (for example an hour), the token will no longer be valid and any requests from microservices to the OAuth authorization server will result in it being refused. The microservice will report back to the user advising it cannot complete the operation.

Username and password credentials do not have this dynamic level of access granularity or time limit restrictions leading to tokens being much more secure and flexible.

The tokens are small files which must be validated by the OAuth server and one of the most used formats is JWT (JSON Web Tokens). As well as providing a convenient method of RESTful data exchange, they provide a method of securing and validating the data through encryption.

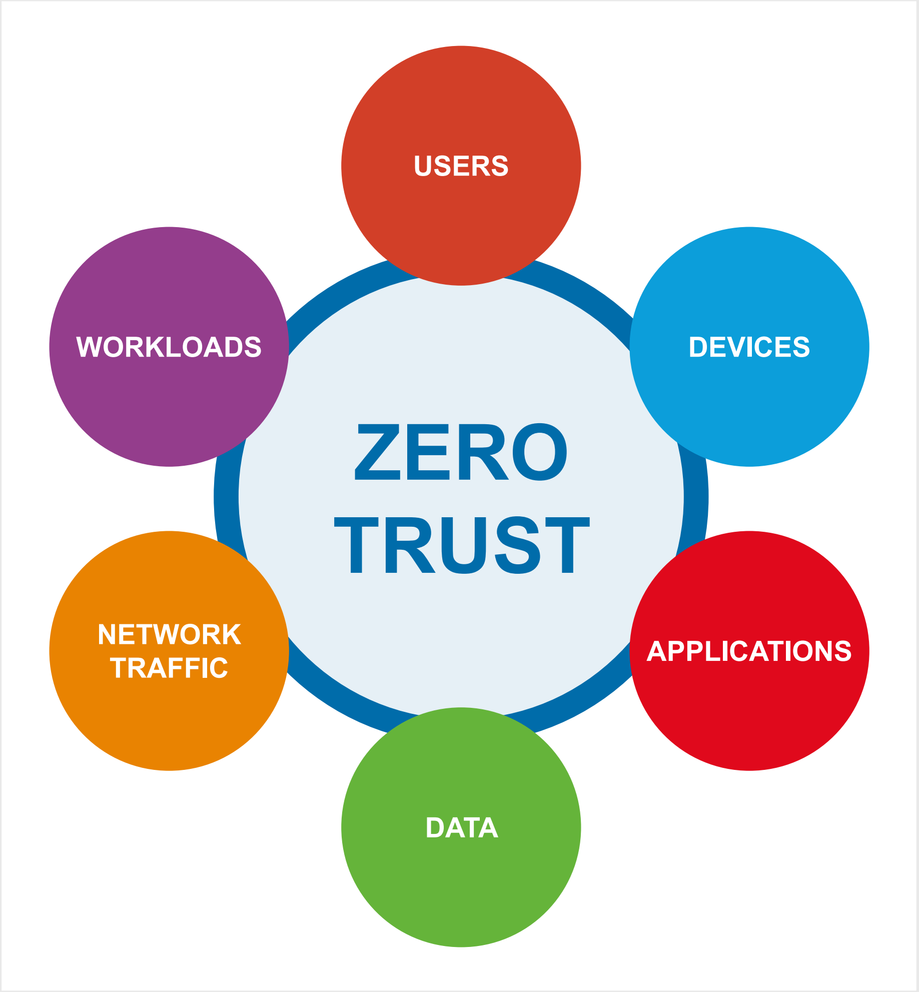


Figure 3 - Zero Trust questions and validates all resource processes within the IT infrastructure.

### Zero Trust

Security of modern cloud systems falls under the heading of "cyber security". This specifically details the protection of computer systems from unauthorized or malicious users to prevent damage to the system or data theft. And one of the most prominent methods of achieving this is through Zero Trust, an architectural model that has at its core the concept of "never trust, always verify".

Work conducted by US cybersecurity researchers at NIST (National Institute of Standards and Technology) and NCCoE (National Cybersecurity Center of Excellence) resulted in the publication "SP 800-207, Zero Trust Architecture". The idea behind Zero Trust is that the traditional model of maintaining a secure perimeter for the network is replaced by assuming that any data exchange, storage, or processing point is a potential threat. This may be through the internet, Wi-Fi, USB drive, or even malicious software masquerading as a legitimate product. Everything and everyone presented to the network must be continuously verified.

For example, a user logging onto a workstation must be validated, preferably through a centralized user verification system such as Okta or Azure Active Directory (AD). The AD then sends a message to the OAuth authorization server to create a token, and this token is used by the microservice to check with the OAuth server that it is authorized to execute the process as well as defining its scope. Whenever the user tries to access any point within the infrastructure, the same OAuth token is used to verify their access thus greatly improving security.

The fundamental difference between perimeter and zero trust models is that the perimeter method tries to build a wall around the infrastructure to keep hostile actors out, however, the zero-trust method assumes we can never achieve this, and hostile actors are everywhere, not just where we would like them to be. By verifying each access using the OAuth token, every step in the workflow is secured.

The OAuth token is unique for every user session and not only maintains read-write-execute rights, but also has a time limit placed on it. As each process execution, data transfer and storage function are validated by the microservice against the OAuth authorization server, any suspicious behavior can be stopped by terminating the token, thus stopping any further processing. This would not be achievable in the traditional perimeter model as once the user is within the perimeter they are difficult to stop.

Another major benefit of the zero-trust model is that every data access and storage process is validated against a centralized OAuth server, and this creates a wealth of metadata. Should a system be attacked, then a forensic analysis of the problem can be very quickly established so that the problem can be dealt with quickly and any security vulnerabilities rectified.

## Collaboration

Security isn't just about working in isolation, instead, it relies on international collaboration. In the same way police services throughout the world collaborate to catch criminals crossing borders, then cybersecurity organizations collaborate to stop cybercriminals attacking networks. One example of this is the organization CVE (Common Vulnerability Exposures) whose mission is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

A vulnerability can be thought of as a mistake in the code that gives an attacker direct access to the network or system. This might be a vulnerability that allows an attacker to pose as a superuser thus giving them unauthorized access to the data.

Software developers use libraries all the time and subscribing to organizations such as CVE allow them to constantly check their own code as well as receive notifications of vulnerabilities in the libraries they are using. Cybersecurity collaboration is a fundamental requirement for keeping cybercriminals at bay and developers have a moral obligation (and in some countries a legal obligation) to notify organizations committed to combatting cybersecurity should they find a vulnerability in their code or others.

## Conclusion

Maintaining high levels of cybersecurity is not just a technical challenge but applies to users of the system too. From the CEO down through the whole organization, keeping systems secure is the responsibility of everybody. This is particularly important when working with cloud systems and microservices as they are always online and so need to be secure at all times.

Broadcasters need to be certain that the software they are introducing into their infrastructures, whether on-prem or public cloud is secure. And the best way of achieving this is to check the provenance of the code being introduced.

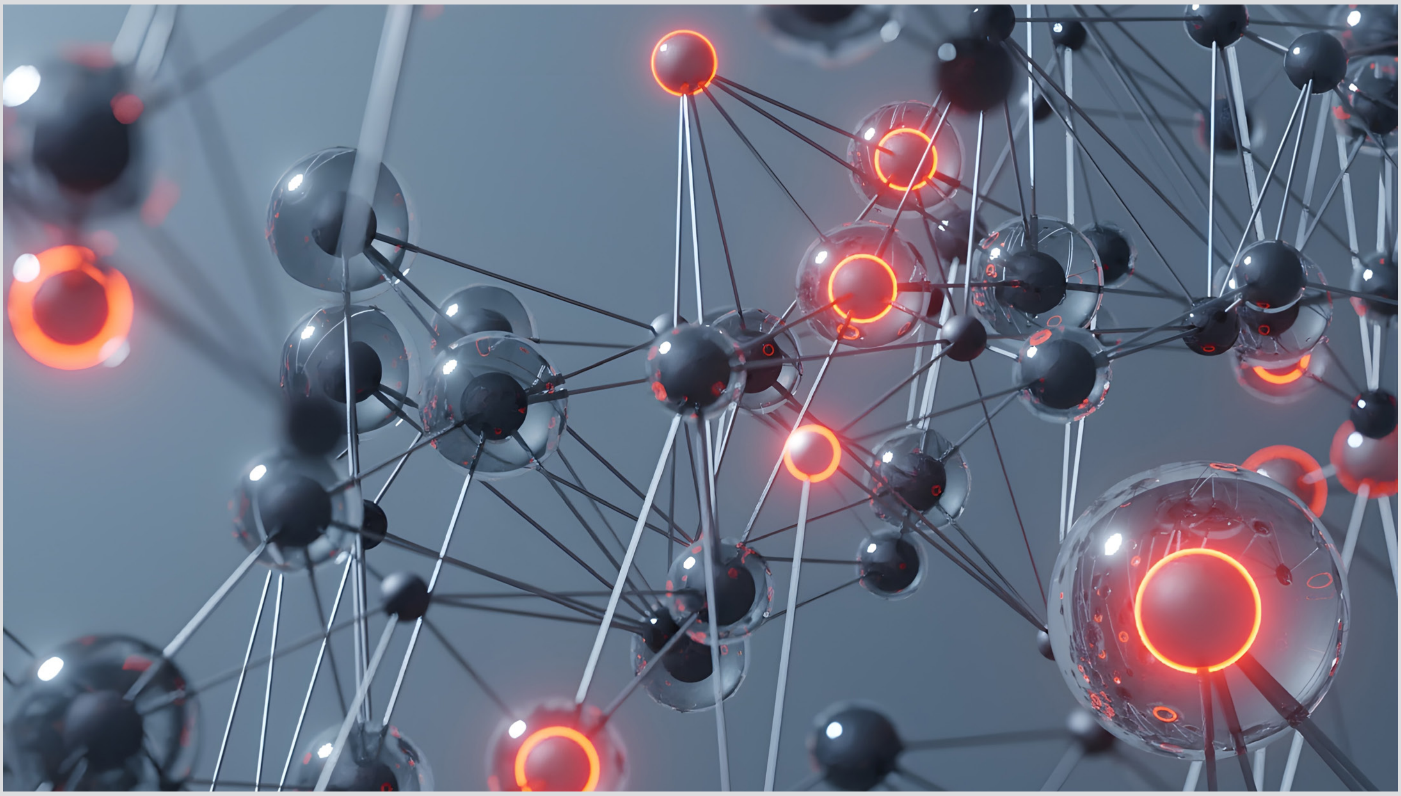
Zero-trust infrastructures are key to keeping broadcaster systems secure and high value media assets safe.

# The Sponsors Perspective

## Trust No One

By Chris Merrill

There are many philosophies out there about who and when to trust. When it comes to securing high value assets, you really can't be too careful.



SaaS processing and storage offers great benefits to media companies:

- Remote production workflows become easier and more affordable.
- Collaborative work among geographically separated teams is simplified.
- Resources scale to match the immediate demand.
- Multi-stage processes are automated and consolidated.
- Productions costs are easily matched to asset revenue.
- The list goes on…

But any member of the digital community has heard worrisome stories about distributed denial-of-service (DDoS) attacks, data breaches, and other digital security issues.

And while malicious actors make the headlines, researchers from Stanford University and the security firm Tessian found that approximately 88% of all data breaches are actually caused by an employee error[1]. So how do we ensure that valuable data remains secure?

The answer is Zero Trust. Zero Trust is not a manifestation of extreme paranoia. Rather, Zero Trust is a strategic approach to cybersecurity that secures an organization by eliminating the assumption that any user that is inside the network firewall should have free access. A Zero Trust strategy continuously validates every stage of a digital interaction.

Rooted in the principle of "never trust, always verify," Zero Trust is designed to protect modern environments and enable digital transformation by using strong authentication methods, leveraging network segmentation, preventing lateral movement to other hosts or applications, providing Layer 7 (application layer) threat prevention, and simplifying granular, "least access" policies.

Because Grass Valley's AMPP is a SaaS solution that runs on any infrastructure, it can offer URLs to the public internet. Here's what we've learned about keeping the production platform secure.

### Require Every User To Sign On With Their Own Credentials

For small installations, user credentials can be configured inside the AMPP Identity service, where users have their passwords secured in a salted one-way hash. AMPP Identity can also be connected to a customer's Identity Provider, to securely delegate authentication and authorization. An external Identity Provider can perform multifactor authentication if required.

### Encrypt All Traffic

Following AMWA security recommendations for NMOS, AMPP only supports HTTPS encrypted traffic. This ensures that no man-in-the-middle attacks are possible while communicating with edge devices or user interfaces.

### Secure Every Call To Every URL

A typical monolithic "lift and shift" development uses a simple password. Once the initial password is compromised then the entire system is vulnerable. AMPP keeps the simplicity of a Single Sign On with assignable roles and responsibilities for each user – often through an external Active Directory – but takes security a step farther with a modern microservice architecture.

Each microservice has its own URL. These URLs are assigned to specific units that form specific tasks. Authorization checks are required when exchanging information between each of these units, so even if one unit is compromised there is no simple means of spreading out to other units.

All traffic inside the platform requires URLs to carry an "OAuth2 OpenID Connect (OIDC) JSON Web Token." That complex statement strings together three different security schemas from different providers in multilayered security. It means:

- OAuth2: The tokens prove the source of the request is from an authorized user without providing the user's password.

- OpenID Connect (OIDC): The identity of the user making the request can be authenticated against an external source.

- JSON Web Token (JWT): The token used for the request is digitally signed by a cryptographically secure signature to ensure nothing has been tampered with.

All this exchanging and validating of information takes place at speeds that never impact the real-time performance of the system.

### Limit Duration Of Credentials

Each time AMPP users log in, their identity is issued a timeboxed JWT which is no longer valid on expiration. The software the users interact with must provide their secure JWT to each RESTful endpoint they call. Because these JWTs are time constrained, it narrows the window opportunity for access to the system.

Because JWTs are constantly refreshed by all client-side libraries, if an admin changes the access rights for an individual, the new JWT issued will reflect the new access status.

### Encrypt All Data Stores

Whether your data is stored on-prem or in the cloud, it needs to be protected while at rest as much as when it is being transported. Hence, all data stores in AMPP encrypt their data before storing, so that even if the content of these stores is compromised, the data is useless to the attacker, who has no ability to decrypt the content.

[1] https://www.tessian.com/research/the-psychology-of-human-error/

### Secure Workloads

It's not just humans that need authentication. All AMPP Workloads are issued with Client Credential Keys that limit their access to all APIs. In the same way that a human user needs to provide credentials to be authenticated and authorized, so do all software components. Client Credential Keys can be managed from within the AMPP Identity user interface.

### Audit, Audit, Audit

Just as malicious actors never stop trying to enter the system, AMPP never stops looking for weaknesses to strengthen. This constant process is part of the SOC 2 certification. Grass Valley has gone through a rigorous evaluation by a trusted third party to be accredited with SOC 2 compliance.

By implementing the latest in technology, AMPP conforms to the best practices of the IT industry. AMPP provides a reliable, secure work environment for creating valuable content. We're not asking you to trust us. We're asking you to put it to the test.

Chris Merrill.

THE BROADCAST BRIDGE
Connecting IT to Broadcast

Themed Content Collection

EG
ESSENTIAL GUIDES

MEDIA

ARTICLES

For hundreds more high quality original articles and Essential Guides like this please visit:

thebroadcastbridge.com

02/2023

Supported by

Grass Valley
WE LOVE LIVE