IT Security in Broadcast Environments

By Jay Bergman
Jan 11, 2022

Introduction

Media content in broadcast and streaming environments is extremely valuable. This is particularly true for first-run broadcast shows and recently released movies. Content and its associated metadata are intellectual property (IP) and create great value for many companies. In the past, content has been stollen and publicly released and content has been locked from use by ransomware attacks. Unsecured communication links can also allow passwords to fall into unauthorized hands and disrupt control channels that move and playout the media content. Security should be integrated into each part of the system and should not be an afterthought. It is an easy step to bypass in the design and implementation stages which will save time and money but the potential ramifications on future losses are significant.

This paper is not a "how to" guide for implementing the security methods but will provide an overview of the important considerations and the tools available to provide secure access within the Zero Trust model. This includes access and communications within and outside of the secure system. System designers, developers and integrators and especially management need to be aware of the consequences of not deploying the necessary security. Broadcast systems can no longer be treated as trusted closed systems as every communication must be secured.

How We Arrived Here

Up until the late 1990s, broadcast equipment were treated as appliances. They connected to other equipment via audio, video and serial control cables and did not have accessible operating systems (OS). Security issues consisted of keeping unregistered personnel out of technical operations and equipment areas. From the late 1990s forward, broadcast equipment began utilizing standard operating systems such as Windows and Linux, and had Ethernet ports in order to connect to or communicate with other equipment.
These connections were made via Ethernet switches on local area networks (LANs) and as these systems grew, they also connected via routers and firewalls over wide area networks (WANs). Well-designed and maintained firewalls served the function of keeping unwanted traffic out of critical networks as these were considered closed systems.

With the introduction of Windows-based applications on the network and the need to patch these applications, a path for malware was introduced. As the network was usually locked down via firewalls, the most common method to perform application and OS patching was for either a vendor or a local engineer to utilize a flash drive. This in turn set off the requirement for more frequent OS patches and the use of anti-malware products. More firewall ports and paths needed to be opened for these products which exposed the previously closed network to the outside.

There are two types of broadcast environments. The first is the environment that is transitioning from the traditional standalone network to one where security layers are being introduced over time. The second is the one built from the ground up with security as a main design consideration. A transitioning model is more difficult to secure as it has legacy products and designs which may not meet the current security requirements. One of these requirements which would be established in a new system is the concept of Zero Trust where every device, application, user and data flow is treated as untrusted until verified.

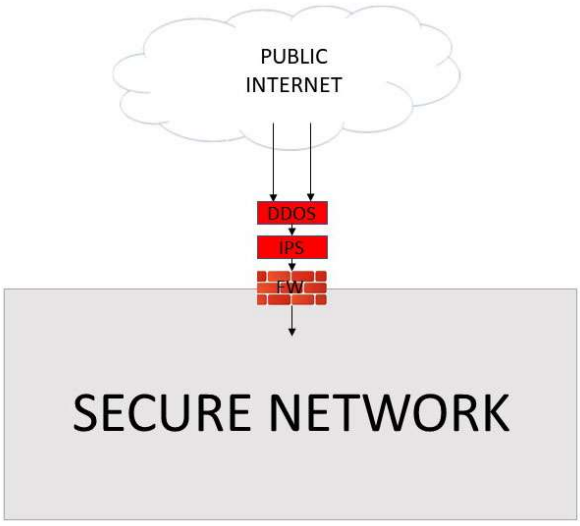
Securing the Network from the Outside

Firewalls

When designing our secure network, we first need to protect it from outside intruders. The firewall is the first line of defense. If there was no need for contact with outside entities or providing vendor or employee access from outside of the system, then there would be no need to protect the system from outside access. We know that broadcasters have to move media to/from content creators and to content consumers. We also know that employees and trusted vendors require access to the system in order to maintain and support the vital hardware and software. Firewalls provide this restricted access with Access Control Lists (ACLs) which act like filters allowing specific sources to communicate with specific destinations using specific ports. If a communication does not match an ACL, it is not allowed access. These ACLs can be setup to be very specific or very wide. The more specific they are, the more the secure the system will be.

Note: Different applications and communication protocols use different virtual access doors called ports. Adding these to ACLs provides a higher level of security.

FIGURE-1 – SECURING THE INTERNAL NETWORK



Firewalls can be purchased as appliances or can be as inexpensive as using the IP Tables utility on a Linux server. Each has the same result – Utilizing ACLs to allow only the required

communication traffic to pass. Appliances provide a more simplified user interface which shields the user from the more complicated ACLs under the hood. The user of the IP Tables utility would be managing all of the ACL filters manually. While the Linux server path will seem like a financially better choice, be aware that it requires a staff that is very well versed in Linux and ACLs and there will be no vender to call upon when support is required.

Inside and Outside Threats

Depending upon the size of the network and the business units involved, a firewall can be setup to protect the system from outside entities as well as from different internal business units. Protecting business units from each other is not usually setup for trust issues but rather to contain malevolent actors and applications within smaller networks and to protect the larger system as a whole.

In-line Inspection

While the simple firewall functionality restricts communications to allowed paths, an Intrusion Detection System (IDS) will alert system administrators of suspicious activity by actively scanning the packets that pass through it as it looks for specific signatures. An Intrusion Protection System (IPS) will not only provide alerts, but will take actions to stop the activity. The IDS/IPS can be a standalone appliance, it can be integrated with the firewall or it can be part of a host workstation or server endpoint anti-malware product. Utilizing this on or in front of a firewall will stop the intrusion before it gets into the secure network. Depending upon the amount of content actively flowing through the firewall, the IPS/IDS can be expensive, can require high processing power and, if not properly purchased, can be a throughput bottleneck. Note that these are subscription-based products that are constantly updating with new malware definitions.

Denial of Service Attacks

Denial of Service (DoS) attacks are attacks that originate from host computers outside of the trusted network. If an attack is coordinated from multiple computers, it is called a Distributed Denial of Service (DDoS) attack. Their goal is to exhaust the resources available to a network, application or service so that legitimate users are denied access. DDoS protection can be integrated into certain firewalls, IPSs and dedicated appliances. As firewalls and IPSs are optimized to perform their main tasks, they may be overwhelmed with a large DDoS attack. For critical systems, it may be prudent to utilize a dedicated DDoS hardware appliance which would reside outside of the main firewall. It is another critical decision that requires product research and consideration of business requirements.

Resources

Most large companies already have the resources discussed above protecting the internal network(s) from the Internet. A coordinated effort should be undertaken in order to make sure that these resources are scaled for any additional planned requirements that will be added to the broadcast systems. For example, new workflows which move or stream files

to/from outside sources or destinations will require additional bandwidth. Existing firewalls and security appliances may need to be upgraded or replaced.

Cloud

Utilizing cloud-based systems is a large topic which will not be detailed in this paper. Using cloud services such as Software-as-a-Service (SaaS) requires a much different architecture than moving entire workflows to the cloud. Bandwidth, security and disaster recovery are just some of the considerations which will differ. Configuring, managing and maintaining systems in the cloud also requires different skill sets to be acquired by the technical staff.

Securing the Network from the Inside

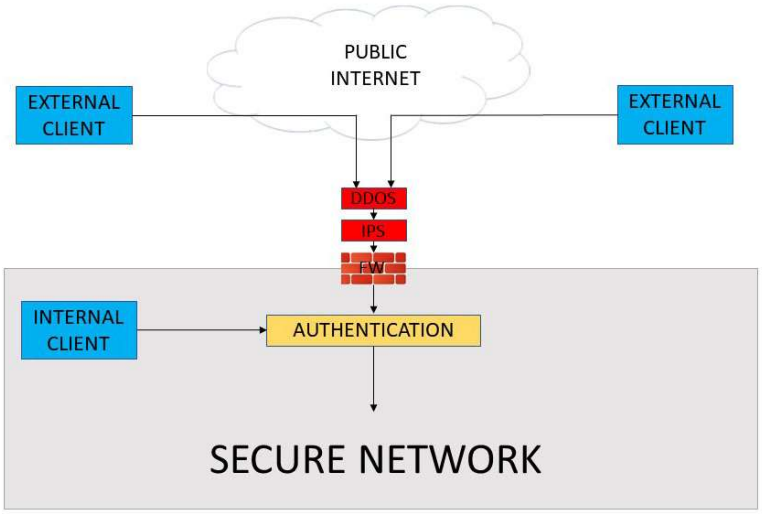
We have reviewed the method where security is enforced by IP address. This was also the traditional method used for static systems where there were clear, fully trusted network boundaries. In today's modern dynamic systems, we can add low-trust cloud and multi-cloud environments with unknown network perimeters across the clouds. Once inside the network, securing the internal infrastructure of a network requires a layered approach. This consists of adding a set of security layers enforced by identity. Once again, it begins with Zero Trust but also it requires constant vigilance to maintain a clean system. Some of the approaches can be static and automated, but many require continuous attention and maintenance. AAA refers to a standard-based framework – Authentication, Authorization and Accounting which mediates network access. These will be discussed in this section in addition to Secure Transport protocols and Secure Storage.

Authentication

Authentication is the process of determining that someone or something is who they are supposed to be, utilizing credentials such as username, password and/or other forms of identification. Entities called Identity Management and Directory Services are database applications that store the users, passwords and computer accounts, provide authentication, group and user management and policy administration among other tasks and share the information with other entities on the network. Popular examples are Active Directory (AD) for Microsoft Windows domain networks and Identity Management (IdM) for Red Hat Enterprise Linux (RHEL).

Examples of other directory services are OpenLDAP and Apache Directory Server. AD and IdM support the Lightweight Directory Access Protocol (LDAP) which is an open and cross-platform protocol used for directory services authentication. This allows for sharing authentication information across different services. IdM also has native support for AD.

FIGURE-2 – SECURING INTERNAL INFRASTRUCTURE WITH AUTHENTICATION



Many enterprise grade network devices such as Ethernet switches and routers do not usually interface directly with directory services nor do they support LDAP. They usually support a legacy protocol called Remote Authentication Dial-In User Service (RADIUS) which also allows for passing authentication requests to an identity management system such as AD. A RADIUS server is, in effect, a translator supporting the AAA framework that allows these devices to communicate with the identity management system when they do not natively speak the same protocol. Switches and routers that support the AAA framework can be implemented using a local database of the device either as a primary or a backup. Usage examples include local engineers authenticating access to network devices, users accessing WiFi via WAPs and WLAN controllers communicating with AD for centralized identity management. RADIUS servers use shared secret keys which are never sent over the network.

DIAMETER is a new enhanced version of RADIUS. The main difference is that RADIUS is a connectionless protocol utilizing User Datagram Protocol (UDP), DIAMETER is connection oriented using Transmission Control Protocol (TCP) which makes it more reliable and supports the AAA**S** framework with the additional **S**ecure Transport.

Multi-Factor Authentication

The use of Passwords is the original and most widely used method of authentication but it is also the easiest to crack. This has been enhanced with Multi-Factor Authentication (MFA). MFA encompasses two and three-factor authentication techniques. The concept is that each additional factor or evidence that is required for authentication adds another level of security, however, the more added factors, the more onerous it becomes for the user to access the system.

Multi-factors usually include one or more combinations of the following:

- Something the user knows – Password, PIN or Secret Question

- Something the user has – Connected (Card Reader, USB Stick, Near-Field Communication (NFC), Radio-Frequency Identification (RFID) and Bluetooth Devices) & Disconnected Security Tokens and SMS Text-based Tokens. Tokens are better than passwords or PINs as a dynamic (different) code is generated each time a user logs in. While it may be the simplest method to implement as almost everyone carries cell phones, SMS tokens are not the most secure authentication method as cell phones can be compromised.
- Something the user is – Biometrics (Fingerprint, Iris, Face or Voice)

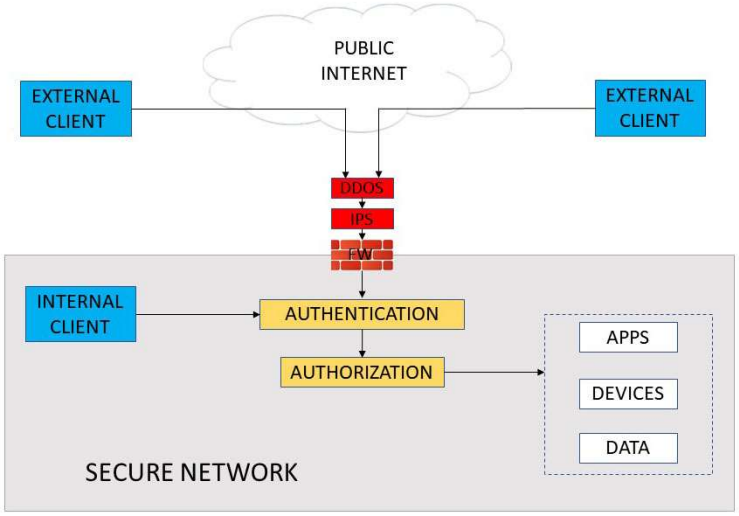
Takeaway 1 – It is not realistic to assume that one service can be utilized to provide authentication for all devices in a large system. Using common protocols is a method of bridging services.

Takeaway 2 – The use of multi-factor authentication is highly recommended for those entities that have valuable information or systems that cannot afford to be compromised and taken off-line.

Authorization

After authentication is successful, authorization is the process of determining what resources the user has access rights to and what operations the user is allowed to perform. Assigning users to groups within identity management services is a simple method of providing authorization to users or service accounts. The groups should be configured with the least privileges so that each user can perform only their required tasks.

FIGURE-3 – ADDITION OF AUTHORIZATION LAYER



When configuring accounts for new users or applications, do not allow admins or vendors to take the easy path and provide superuser accounts such as root in Linux and membership to Administrators, Domain Admins and Enterprise Admin groups in Windows to users or service accounts as any malware or hacker gaining access to these accounts can cause great

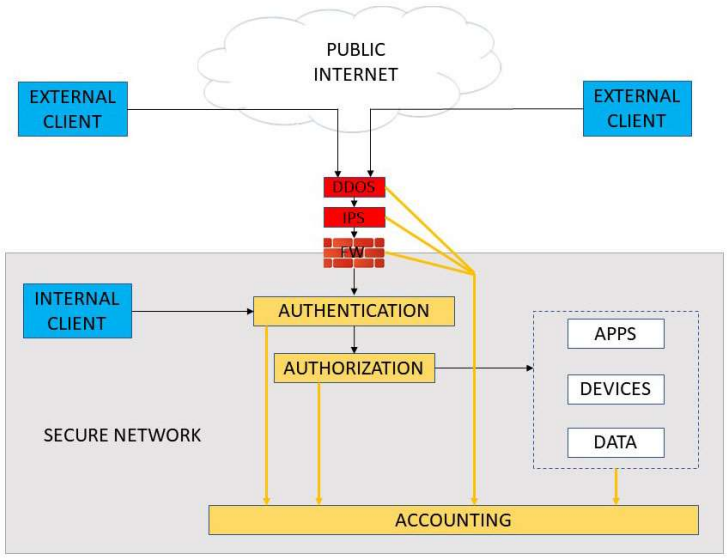
damage. This is an easy method to get a system functioning quickly but it bypasses all security required to keep the system safe.

Takeaway – Configuring and maintaining accurate and up-to-date user authorizations in a large system can be a daunting administrative process. Assigning users and accounts to groups and utilizing the principle of Least Privilege can help simplify the process.

Accounting

The last part of the AAA framework is Accounting. It provides a means for monitoring and capturing the activities of a system, a user or an account while it accesses network or system resources. It is achieved by logging session statistics and usage information. The accounting data can be used for auditing, billing, application, network and database real-time monitoring, debugging and trend analysis and planning and resource utilization. Log data can be very useful in capturing user actions and resolving timing and performance issues in broadcast environments. As seen in the figure below, applications and devices provide accounting data.

FIGURE-4 – ADDITION OF ACCOUNTING LAYER



➢ Logging

Logging is often not on the top of the list when it comes to designing and pricing a new system. Most applications, appliances and security systems provide logs, often in user configurable levels. For example, most broadcast automation applications have a verbose logging level which logs every user and system activity. The more logs that are stored, the more information is available when required but it comes with a price. Logs generate network traffic, they take storage which must be managed and depending upon the utilized storage system, there can be an additional cost tied to the storage amount, search and analytics processing power and the amount of daily captured data.

➢ Log Storage

Users often have a choice to have each application or device capture its own logs locally or to unify the logs in a common storage solution. A unified approach takes more time to configure but creates a more simplified process of accessing and correlating the data as well as providing the ability to layer business intelligence or machine learning on top of the data. Ideally logs should be buffered locally to avoid network bottlenecks and failures and then transferred to common storage.

Syslog is a common log format which is used by most applications, devices and log storage devices. Existing utilities can be used to convert from proprietary log formats to the Syslog format. When setting up a data log, it is very important to utilize a common data format for the payload from each log generator in order to simplify setting up the storage ingest and reporting. Include time-stamps for every log with the same time-zone and daylight savings time format regardless of origination.

The amount of time logs should be kept depends upon several factors:

- Compliance Regulations – Often governmental or corporate regulations will determine the amount of time logs need to be held.
- Internal Requirements – Logs often no longer serve a useful purpose after a period of time.
- Storage Pricing – While the price of storage is always decreasing, it still has an ongoing cost. Logs can grow very large, very quickly.

➢ Logging Tools

Once the log storage requirements have been determined, logs can be setup to automatically delete after a given amount of time. This does not have to be a manual process until requirements change. Log data can also be configured to move to lower-priced archive storage tiers over time. Archives are readily available and simple to configure in cloud-based environments.

Note: It is recommended that each user has their own account and shared accounts be kept to a minimum. This will allow for user-related system problems to be more easily diagnosed and resolved as logs can be correlated with the user activity.

Maintaining a Secure Environment

The AAA framework is meant to provide a secure network by controlling access to the environment. As we are discussing a layered approach, we also need to assume that this secure network may be breached and we therefore need to maintain a clean and controlled environment. Systems need to be in place in order to keep appliance firmware, application software and operating systems (OS) patched with the latest approved releases. Systems also need to be deployed which scan for abnormalities and unauthorized or out of date applications or processes.

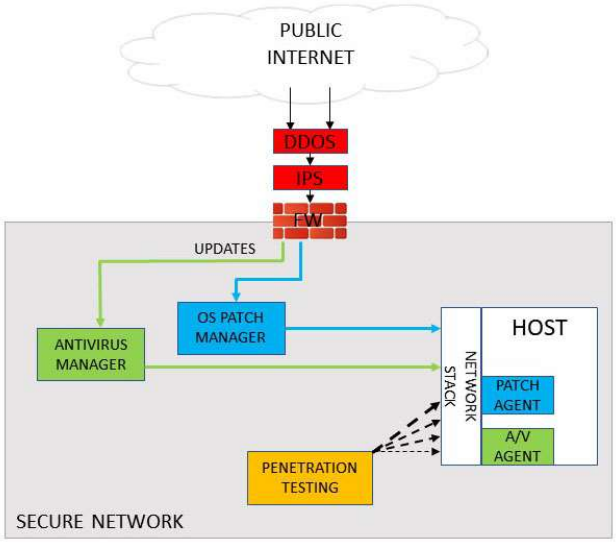
Patching

Patching of OSs, applications and appliances is a contentious topic. We all know that maintaining the system with the latest security patches is paramount to the security of the system. We also know that the latest patches are not always free of bugs or compatible with the other system components. The ideal procedure when deploying new OS patches is to first have the vendor(s) whose application is potentially affected by this patch – approve the patches. An example would have a vendor whose software runs on Windows approve the latest patches. Once approved, the patches should either be tested on a dedicated test system or on a small set of deployed applications. For appliances such as Ethernet switches or firewalls, a backup unit should be put into service with the new firmware which can be quickly switched back in case of problems. A system can also be taken out of service, if possible, in order to run the required tests but care must be taken in order to properly simulate the running environment such as work-loads, work-flows and timely (frame accurate) communications. Detailed test procedures must be generated and followed every time a new patch is deployed.

Automated OS or application patching is not recommended in most environments. Testing as discussed above gets bypassed and applications and appliances will go off-line during a reboot and reload process. Many applications such as real-time broadcast automation are finely tuned to the OS and network stack and may be adversely affected by small changes. Also beware when configuring OS patch management systems as they have the ability to download, install and/or reboot hosts which require security patches. These systems can install and update OSs across entire networks at the same time.

Obsolete or unsupported applications pose another serious problem. They will not be updated with required security or bug patches and may not be compatible with new OSs. Old unsupported OSs will need to be maintained but will add to security risks. While it may be a financial hardship to replace, maintaining old applications or appliances will end up costing the company more in the long run.

FIGURE-5 – HOST PATCHING AND SCANNING



Scanning

We need some method of monitoring a system for unauthorized or dangerous actors. Manual or automatic scanning can perform this task. Scanning applications can look for viruses, malware, OS, Application and Appliance security patches, open ports and unauthorized applications and processes. An agent is usually installed on the host which reports back to the manager. There are, however, issues to be aware of on this topic.

➢ Antivirus Scanners

It is not recommended to have each host acquire the latest virus and malware definitions from the vendor website. Usually this requires a range of IP addresses to be opened on the firewall for each host. A more ideal solution is to have an Antivirus deployment manager on each subnet, each receiving the definition updates and communicating with each internal host. Only the deployment managers require firewall ports to be opened. Also be aware that some antivirus vendors store host data on their servers. If this poses a security risk, then it must be addressed. The act of antivirus scanning on local drives can also have an adverse effect on file access and on the network stack for real-time broadcast automation systems.

➢ Host Scanners

Applications that perform security checks such as penetration testing by scanning for open ports and for OS patches from outside of the host can also have an adverse effect on the network stack for real-time broadcast automation systems. The host will utilize internal resources in response to the scans which may be detrimental to the active applications and services.

Secure Storage and Transport

Designing and maintaining a secure network includes protecting the data and communications from malevolent actors that may have penetrated the system's defenses. There are several methods of securing the network, each of which involves encryption. The following fundamental terminology for encryption and hashing will help explain the frameworks and protocols utilized to ensure secure storage and transport.

➢ Encryption

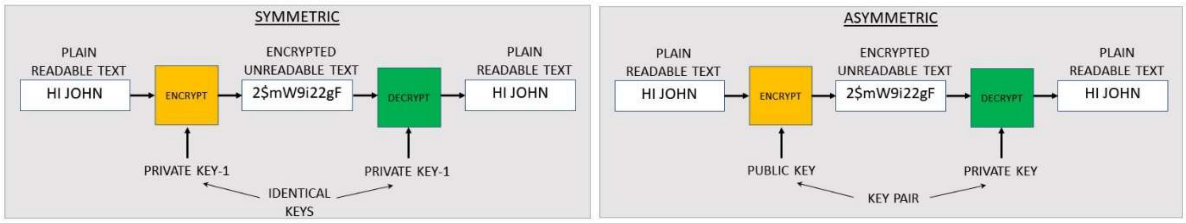
Encryption is the process of converting readable (plaintext) data into unreadable (ciphertext) data in order to keep it private. There are three states of encryption:

- Encryption at rest – Protects the data from a system compromise while stored. Typically uses the Advanced Encryption Standard (AES) encryption algorithm
- Encryption in transit – Protects data as it moves between two hosts or services. The data is encrypted at the source, the endpoints are verified and the data is decrypted at the destination. Frameworks such as SSL, TLS, PKI and Kerberos, which will be discussed later on, are used for this process.
- Encryption in use – Data residing in application memory is encrypted in order to prevent it from being compromised while it is being processed.

➢ Encryption Keys

Encryption keys are random strings of numbers and letters used to scramble and unscramble data. The longer the string is, the more difficult it is to break the encryption code and unscramble the data.

FIGURE-6 – SYMMETRIC & ASYMMETRIC KEY ENCRYPTION



- Symmetric keys – The same private key is used for encryption and decryption.
- Asymmetric keys – A public key known by everyone and a private key known only by one party are used for a secure conversation. Only these two keys can encrypt and decrypt each other's data. The holders of the public key know that their data can only be read by the holder of the private key.

➢ Data Hash

Hashing is an algorithm that takes a single input (message) and produces a fixed size number called a hash or message digest. The sender and the receiver agree on a hashing algorithm. They will each, therefore, obtain the same hash value when used on same message which provides an integrity check for the receiver. The following is an example of its usage:

Typical Hashing Workflow

1. Client browser and server agree on a hashing algorithm
2. Server uses hashing algorithm on data packet to produce a message digest
3. Server sends data packet and message digest to client
4. Client uses the same hashing algorithm on the data to produce a message digest
5. Client compares the message digest sent by the server with the one it generated
6. If both are identical, the data has passed the integrity check

➢ Secure Storage

Data that is considered to be valuable such as user or employee information or broadcast media can be stored securely by encrypting it. The encryption and decryption are usually performed by the host computer as the data is written to or read from the storage. Any entity that retrieves this data without the encryption key will not be able to read the data. Depending upon the application and/or the storage appliance, encrypting may be as simple as checking a box when configuring the storage. The same can be said for some cloud storage providers and some will encrypt by default. The data may also be hashed. This will serve as an integrity check verifying that the data has not been changed or corrupted since the time it was written to the storage.

➢ Secure Communications

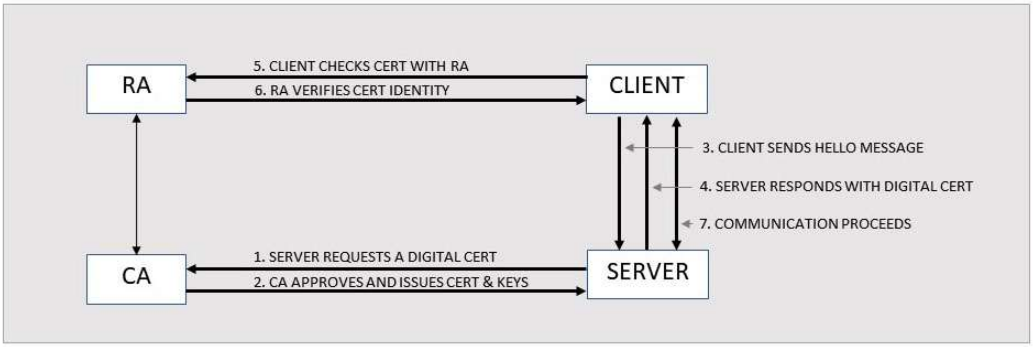
Data must be secured between a sender and a receiver whether one or both are inside or outside of the secure network. This also includes to/from storage devices. In addition to encryption for data privacy and hashing for data integrity, a third party is usually involved for authentication of one or both parties. Here we will discuss usage of the popular Public Key Infrastructure (PKI) and the SSL/TLS and Kerberos protocols.

➢ Public Key Infrastructure (PKI)

PKI is a framework for 2-key asymmetric encryption and security that protects communications between clients and servers. It is also a set of roles, policies, processes and procedures required to create, manage, distribute, use, store and revoke digital certificates and private keys.

What does this really mean? We have reviewed the encryption and hashing security features but we have not yet discussed the third-party authentication. In the PKI framework, an outside entity called a Certificate Authority (CA) will become the authenticating party. In the typical use case, a client web browser is attempting to securely communicate with a web server using the secure Hypertext Transfer Protocol Secure (HTTPS) protocol. The figure below details the basic steps of the PKI framework.

FIGURE-7 – BASIC PKI WORKFLOW



Before hosting clients, the entity behind the web server must obtain a certificate from a trusted CA which verifies the authenticity of the website's owner. Once the CA determines that the requesting entity behind the website is its legitimate owner, the entity will receive an X.509 certificate signed by the CA along with a private and a public key. The private key will be securely stored on the server and the public key will be available to clients attempting to communicate with the server. A Registration Authority (RA) will typically perform the entity verifications for the CA.

Basic PKI Workflow (for HTTPS)

1. Client browser contacts HTTPS server (hello) with a list of supported cipher suites and SSL/TLS version
2. Server provides client with a digital certificate and a public key
3. Client attempts to verify certificate with third-party RA

4. Once verified, client begins communications with server by generating a symmetric session key and encrypting it with the server's asymmetric public key
5. Server uses asymmetric private key to decrypt symmetric key which will be used to encrypt communications for the session
6. Using symmetric encryption, they agree to a hashing algorithm
7. Encrypted and hashed communication commences

Note 1: Asymmetric encryption is used to establish a secure session between a client and a server and symmetric encryption is used to exchange data within the secured session. Asymmetric keys are much longer than symmetric keys and take more processing power and time to process. Switching to a symmetric key for session communication is therefore quicker and less processor intensive.

Note 2: A copy of public keys from widely recognized CAs will likely be pre-installed in the user's web browser. Popular Web browsers like Chrome, Firefox, Safari, and Edge/Internet Explorer all come with the certificates of recognized CAs. That means, they already contain copies of those certificate authorities' public keys and can therefore be used for verifying certificates issued/signed by them with RAs.

Practical Considerations with PKI

There is a fee to register with and obtain certificates from a CA. If a company has hundreds or thousands of hosts to register, this could be a significant cost. These hosts can be dynamic such as virtual machines (VMs) or containers spun up and down given demand for a service requiring ephemeral certificates. There is also the consideration of system functionality during an Internet outage where the RA is not available. For these reasons and those of configuration simplicity, many PKI hosts are configured with self-signing certificates. These certificates are signed with their own private keys, not from a CA. They are fast and easy to produce using a local certificate generating application, but they pose a risk as they are not vetted by a third party therefore eliminating the trust step in the process. The security team also will lack visibility as to how many certifications they have, where they are and who owns them.

There is another option to resolve the above issues with self-signed certifications. Many companies will deploy a private, local CA on their internal network. It must be configured as a highly available service and the local technical support must be well trained on its management and support as it will be a critical service for the ongoing operations of the company.

➢ SSL/TLS Protocols

The Secure Socket Layer (SSL) is a cryptographical protocol used to provide secure communications over a computer network. It has been replaced by the Transport Layer Security (TLS) protocol. These are most commonly used as the security layer of the HTTPS protocol for secure websites and use the PKI framework previously discussed, however, TLS also supports other encryption standards that are not part of PKI. Enabling SSL/TLS on servers, VMs or containers can be performed with configuration parameters in the respective web servers, operating systems, hypervisors and container controller managers.

➢ Secure Shell Protocol

The Secure Shell (SSH) is a cryptographical protocol used to provide secure communications over a computer network and was originally designed to replace the unsecure Telnet protocol used to enable a remote command line. With its open architecture, it is used today for many other applications including SSH Secure File Transfer Protocol (SFTP) and configuring an automatic (no password) login to a remote server.

Note: Many legacy industrial and broadcast devices including modular equipment utilize Telnet as the only communication option for configuration. As Telnet by default does not encrypt any data sent over the connection, including passwords, it is feasible to intercept the passwords and is therefore not a recommended option.

➢ Kerberos

Kerberos is another protocol for providing authentication and authorization between clients and servers (or multiple nodes) on unsecure networks. It utilizes a trusted third-party Key Distribution Center (KDS) which authenticates the nodes by keeping a copy of each node's private key. Keys are never sent over the network but a hash of the key is used instead for security. Tickets are assigned by the KDS and used by the nodes to obtain services from other nodes. Although Kerberos is found everywhere in the digital world, it is employed heavily on secure systems that depend on reliable auditing and authentication features. Kerberos is used in Active Directory, NFS, Samba, macOS, Unix and Linux. It uses symmetric-key cryptography but it may also use public-key cryptography during certain phases of authentication.

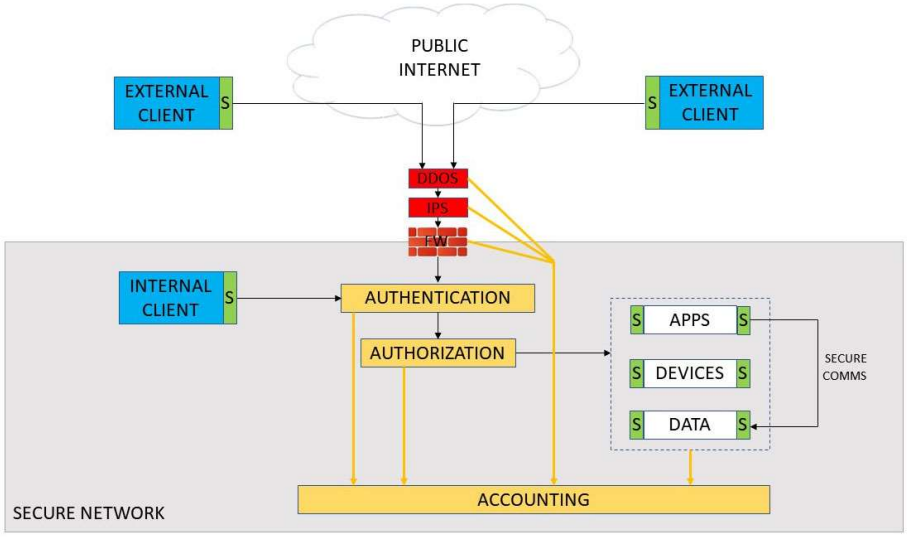
KDS consists of an Authentication Server (AS) and a Ticket Granting Server (TGS)

Basic Kerberos Workflow - Between a client and a service server (SS):

1. Client authenticates upon login with the AS
2. AS forwards the client's username to the KDS which issues a Ticket-Granting Ticket (TGT) encrypted with the TGS secret key and returns it to user's workstation
3. Server has already been registered with the TGS with a Service Principal Name (SPN)
4. Client sends the TGT to the TGS and requests access to the SPN service
5. TGS verifies a valid TGT and verifies the user's authorization to this service
6. TGS issues ticket and session keys to the client
7. Client sends the ticket and service request to the SS
8. SS provides the requested service to the client

Note: Kerberos is a protocol for establishing mutual identity trust, or authentication, for a client and a server, via a trusted third-party, whereas SSL/TLS ensures authentication of the server alone.

FIGURE-8 – SECURE COMMUNICATION LINKS



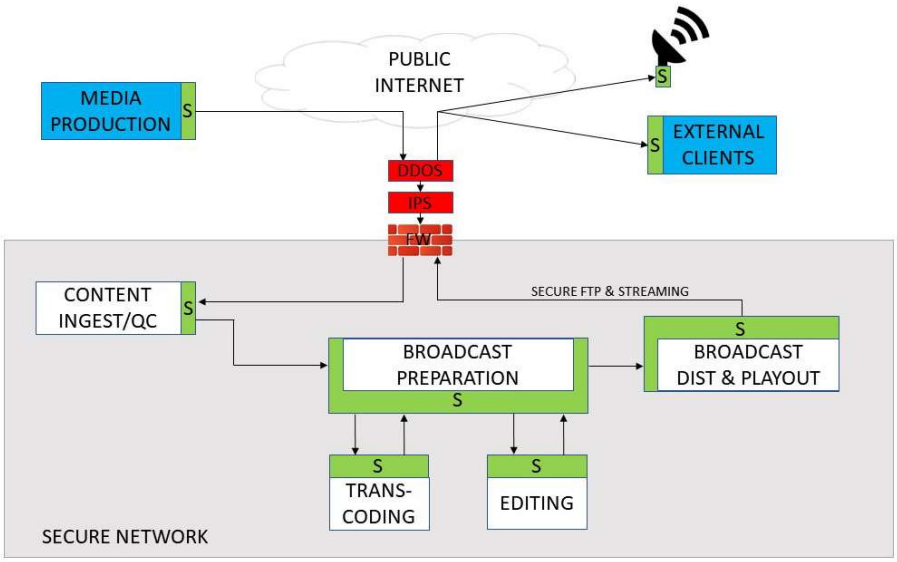
The above diagram shows the addition of the secure communication links between all of the clients, servers and data storage. As previously discussed, this can be achieved using various methods of encryption.

➢ Secure File Transfers

One of the main components of broadcast system workflows is moving files from point to point within or between networks or to/from a cloud provider(s). This can be a part of various use case examples:

Media Production Source –> Broadcast Ingest
Broadcast Preparation <-> Media Processor/Transcoder
Broadcast Preparation <-> Editing System
Broadcast Preparation -> Broadcaster Distribution/Playout

FIGURE-9 – SECURE FILE MOVEMENT IN BROADCAST WORKFLOWS



Traditional legacy systems often utilize the File Transfer Protocol (FTP) to move these files which may include authentication but there is no encryption built into this protocol. If the username and password option is configured, they will be transmitted with clear, unencrypted text. The solution is to use the SSL/TLS extension for FTP known as FTP-SSL (FTPS) which uses TLS encryption. Another acceptable option is to use the SSH File Transfer Protocol (SFTP) which is not compatible with FTPS. Most modern systems will provide these secure protocols as options when configuring file transfers.

FTP utilizes the Transmission Control Protocol (TCP) which provides reliable delivery of data between hosts on a network and over the Internet. It is optimized for accuracy rather than speed. There are proprietary file delivery products that utilize the User Datagram Protocol (UDP), sometimes along with TCP, to provide accelerated file delivery, especially over longer distances. UDP does not utilize handshaking or error correction but instead relies on the application to provide these functions. These products usually also provide encryption for the file transfer process. While these file acceleration applications are more expensive than using the secure FTP variants and the users are often charged by the amount of transferred data, they are very popular with broadcast companies.

We see that there are various methods which protect and enhance the communications between hosts inside and outside of the secure network. One protocol is not necessarily any better than the other. It depends upon, among other things, the application, the OSs and the system architectures. The key is to research the supported vendor application protocols before purchasing the applications and appliances and properly securing all of the communications.

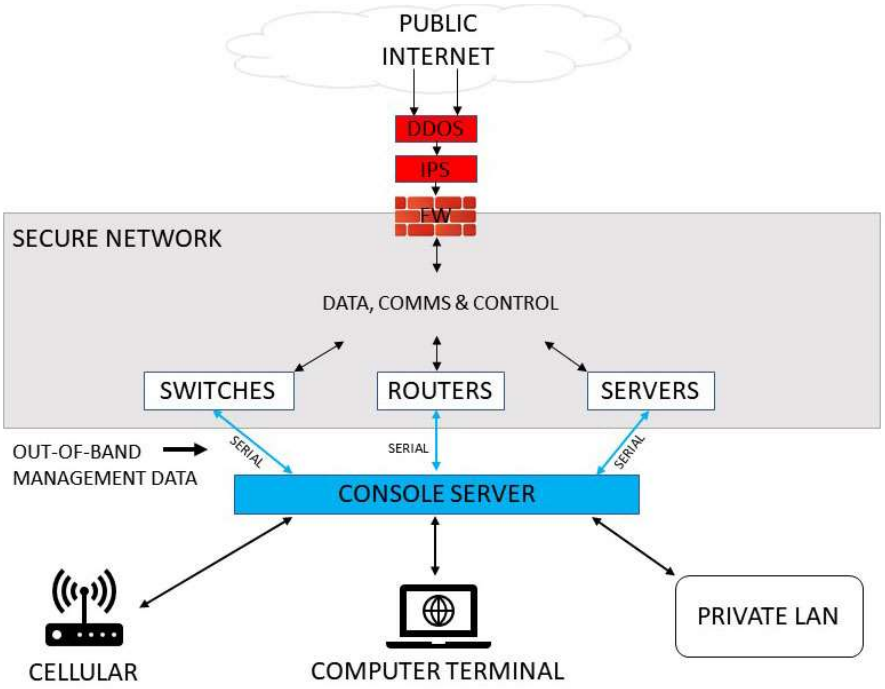
Allowing User, Engineering and Vendor Entry

Application users and support engineers who are already inside the system will have to go through the authentication process before they can utilize the allowed system resources. Entry from outside of the system first requires authentication and then authorized access to one or more specific systems. Popular methods are the use of a Virtual Private Network (VPN) and/or a Bastion host. A VPN extends the secured private network to a user across the unsecure public Internet by using secure encrypted tunneling protocols. The user is essentially securely connected to the LAN as if they were located on-site. A Bastion host is a specialized appliance that sits on the edge of the network, is hardened to attacks and acts as a proxy server typically providing SSH for Linux or Remote Desktop Protocol (RDP) for Windows access to servers on the secure private network. Ideally it would require users to provide multi-factor login, interface to identity management services via LDAP or AD integration and would allow connections only from predetermined whitelisted public IP addresses. Employees usually utilize VPNs while Bastion hosts are usually configured for vendors or outside support personnel.

- **Out-of-Band Management**

Critical equipment such as network switches, routers, servers and storage devices may not be available to remote engineering support should there be a network outage. Most of these devices have serial ports which provide access to a separate computer or laptop configured as a computer terminal. A step up from this would be a device called a console server that can provide local authentication and will allow one or more computers to communicate with multiple serial devices, usually via a Local Area Network (LAN) connection. This LAN can be completely isolated from the devices' network connections for security and for disaster recovery (DR) purposes and, depending on the product, may also be accessed via cellular connections. Utilizing this out-of-band management is much easier than moving a laptop around a data center for engineers and provides a much faster path to resolving network problems remotely. Care must be taken in setting this up in a secure physical location and on a secure network as it will have access to all of the critical resources.

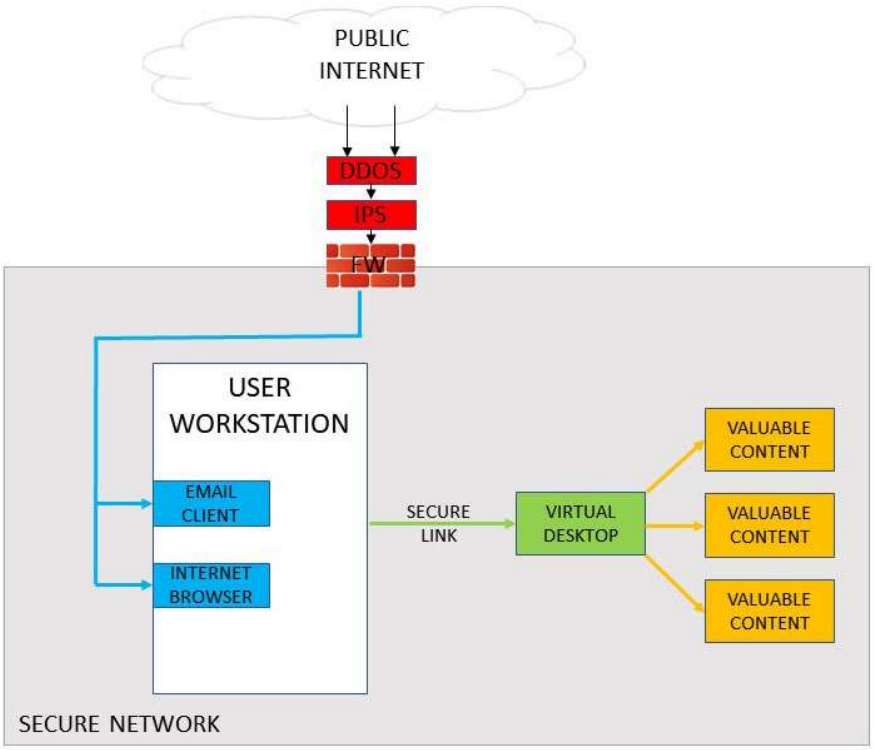
FIGURE-10 – OUT-OF-BAND MANAGEMENT



Integrating Outside-Facing Applications

There are use cases where a user is required to access content on a secure network as well as utilize information or applications that are public (Internet) facing. An example would be access to email from which information is cut and pasted to a secure document. In this case, a solution would provide access to the secure network resource via presentation virtualization. This basically extends the user's keyboard, video monitor and mouse to the remote system configured in a secure manner. There are many supported protocols including Remote Desktop Protocol (RDP) and Virtual Desktop Infrastructure (VDI). There are specialized services in cloud environments. These would include AWS WorkSpaces, Google iTopia Cloud Automation Stack and Azure Windows Virtual Desktop. As long as the infrastructure is appropriately configured, the user will be able to utilize applications on secure and less secure networks without compromising system security. Failure to properly secure environments such as these can result in malware such as ransomware locking up your data.

FIGURE-11 – SECURITY USING VIRTUAL DESKTOP



Protecting Internal Host Identities

When inside users require outside resources or outside users need to access internal secure resources such as web servers, it is often preferrable to hide the identity of the internal hosts. A proxy server acts as a middleman and will be the server exposed to the outside hosts. It can also act as an endpoint for encryption, perform load-balancing activities, cache content to speed up subsequent requests, provide web-content filtering and advanced threat protection. A forward proxy server is used when internal hosts connect to outside resources and a reverse proxy server accepts requests from outside clients on behalf of internal servers.

Other Means of Network Protection

The simplest method of protecting a secure network is training employees to use the best security practices and avoiding social engineering and phishing attacks. Best engineering practices will avoid using wireless protocols such as wifi and Bluetooth within a secure network. For ultimate security a faraday cage would be employed around a datacenter that would protect radio frequency data from escaping the area. Best practices also include disabling unused network device Ethernet and serial ports and tying host MAC addresses to switch ports. When a USB drive is required for file transfer, software updates or device support, an off-line USB malware scanner should be employed.

Active users should regularly be compared to active employee and vendor lists. This will keep potential bad actors from accessing the secure systems. Network activity should also be constantly monitored either manually or via automation in order to detect any suspicious non-routine internal activity.

On-Going Maintenance

It takes a lot of work to keep a system secure. The following are some of the tasks that need to be regularly performed:

1. Key Management/Rotation – PKI certificates expire and need to be replaced on a regular basis as should self-certified certificates. Depending upon the size of the system, this can be a manual process or a PKI automation application can be utilized which will allow for tracking of keys in a single location with automatic rotation. PKI automation reduces errors and ensures that the correct certifications are always requested in a timely manner.
2. Asset Integrity Testing – Once an asset such as a media file is placed on storage, it is susceptible to data corruption which could be hardware-related or the result of tampering by a malicious actor. A checksum or a hash is typically used to verify its fixity to make sure it has remained unchanged since the moment it was written to the storage. An asset can remain in an archive for years before it accessed. It is prudent to use a fixity management system or a workflow which will regularly check the asset fixity by creating a hash with the original algorithm and verifying that it matches the original hash.
3. User List Management – As previously mentioned, an active user list should be checked against the active employee and vendor list in order to keep the system current and maintain a secure environment.
4. Password Management – While nobody enjoys changing their passwords on a regular basis, it must be enforced along with appropriate password requirements for users as well as for service accounts.
5. Application Modernization – The method of developing code as a single monolithic application should be replaced with generating containerized microservices. This allows developers to code and test various application components separately from each other using different languages, if desired. It also allows for caching of communications between microservices and automatic scaling of these services under high load conditions.
6. Obsolete OS Management – Many legacy systems still have active applications which require obsolete operating systems. These applications themselves may be obsolete with no company or developers left to support or rebuilt them. These OSs and applications can be a security risk as there are no security patches available to fix obvious or hidden flaws. Effort must be made to replace these obsolete systems.

Some Technical Considerations

The following are some additional technical considerations to think about when designing a broadcast network.

1. In broadcast systems, local hardware control panels as well as computer workstations with soft panels located in various areas are used to switch sources on equipment such as on-premise video routers and video switchers. In addition to insisting on secure communications, the control system can either be on a layer 2 switching environment or a layer 3 routed network. A routed network would be more convenient to install but control could be lost during firewall or router unscheduled maintenance or failure of these same components. Hybrid cloud environments are unique and are routed environments.
2. The broadcast network can be made more secure by keeping it separate and behind the Corporate IT network. Internet traffic would pass through at least two sets of firewalls and security appliances in order to get to the broadcast network. In cloud environments, the broadcast networks should be kept secure in private VPCs following standard security practices of least privilege in order to protect sensitive content and communications.
3. When using Simple Network Management Protocol (SNMP) for device configuration and monitoring, standardize on SNMPv3 where possible as this employs strong authentication, hashing and data encryption.
4. The control plane communication in the switching and routing environment should be securely configured. Most vendors allow for some form of TLS certificates and private/public key pairs which provide authentication, hashing and encryption to the links.