Wallboard

Technical Specs and Security Overview

Version 1.11

Robert Simon / Co-CEO
rsimon@wallboard.info

Contents

Introduction

This document applies to **Wallboard 1.11** installations and aims to provide a comprehensive technical overview of Wallboard digital signage solution, encompassing both server and client application details, along with in-depth technical specifications and robust security details.

Overview

Wallboard is a comprehensive All-in-One Digital Signage comprising five essential components

- Content Player Applications: Wallboard offers diverse Content Player Applications designed for various platforms, including Android, Windows, BrightSign, LG and Samsung SoC devices.
- Server Application: Wallboard provides a versatile Server Application that supports different deployment options, including cloud-based (SaaS), on-premise, or Hybrid setups.
- Management User Interface: The CMS includes an intuitive Management User Interface that empowers users to create, edit, schedule, and distribute content across their network of screens. This interface offers a user-friendly environment for efficient content management, enabling customization, and ensuring seamless content delivery to the targeted screens.
- Sensor Integrations: Wallboard seamlessly integrates with external sensors, allowing dynamic user interactions with the digital signage system. By incorporating sensor integrations, Wallboard enables real-time data input, trigger-based actions, and personalized content delivery based on specific user interactions, enhancing user engagement and interactivity.
- Edge Server: To address large-scale deployments and provide content availability in low-bandwidth internet locations, Wallboard includes an Edge Server component. This server optimizes content distribution and scaling, improving performance and reducing network latency, particularly in remote or challenging connectivity environments.

With these components, Wallboard offers a comprehensive Digital Signage CMS solution that caters to a diverse range of platforms and requirements. It facilitates efficient content management, seamless content delivery, enhanced user interactions, and scalability for large deployments, ensuring an engaging and dynamic digital signage experience.

The aforementioned components rely on a REST API to facilitate seamless communication among them. This ensures efficient data exchange and interaction between the different components, enabling smooth coordination and integration.

To deliver real-time notifications, Wallboard employs WebSockets, a powerful communication protocol that enables instant data transmission. Leveraging the capabilities of WebSockets, Wallboard ensures timely and efficient delivery of real-time notifications, enhancing the overall user experience.

Security standards

Access Control

Role-Based Access Control: Wallboard implements a hierarchical role-based access control model. Each user is assigned a specific role based on their responsibilities and access requirements. This approach guarantees that users only have access to the functionalities and data necessary to perform their tasks, minimizing the risk of unauthorized access.

Least Privilege Principle: The principle of least privilege is strictly followed, through the team system administrators are able to grant users only the minimum privileges required to perform their tasks effectively, reducing the risk of unauthorized access.

Secure Authentication and Authorization: Wallboard utilizes advanced authentication mechanisms, requiring users to undergo a rigorous authentication process to access the application. Strong password policies are enforced, and there is an option to use multi-factor authentication (MFA) for enhanced security. Authorization checks are thoroughly implemented at every access point to verify user permissions.

Account Lockout Policy: To prevent brute-force attacks, Wallboard implements an account lockout policy. After a certain number of unsuccessful login attempts, user accounts are temporarily locked to mitigate the risk of unauthorized access.

Operational

Regular Security Audits and Code Reviews: Wallboard conducts regular security audits and comprehensive code reviews to identify and address access control vulnerabilities promptly. The development team maintains a secure codebase, swiftly resolving any issues that arise.

Automated Testing and Security Scans: Automated security testing is integrated into Wallboard's CI/CD pipeline. Rigorous security scans and tests proactively detect and address access control weaknesses and other security flaws.

Implement Secure APIs: Wallboard emphasizes securing APIs by using authentication tokens and implementing strict validation and authorization checks on all API requests to prevent unauthorized access.

Least Privilege Principle: Application components and database accounts are granted the least privilege necessary to perform their intended functions. This minimizes the impact of potential injection attacks by limiting the attacker's access.

Incident Response Plan: Wallboard has a well-defined incident response plan in place. This plan outlines the steps to be taken in case of security incidents and ensures a coordinated and swift response to mitigate any potential damage.

Cryptography

Strong Encryption Algorithms: Wallboard utilizes industry-standard and robust encryption algorithms to protect sensitive data at in transit. The selected cryptographic algorithms are well-regarded in the security community and are resistant to known attacks.

Secure Key Management: Proper key management is crucial in cryptographic systems. Wallboard employs secure key generation, storage, and distribution practices to prevent

unauthorized access to encryption keys.

Transport Layer Security (TLS): For secure communication over the internet, Wallboard uses TLS protocols with strong cipher suites. TLS ensures that data exchanged between users and the application remains encrypted and protected from eavesdropping and tampering.

Certificate Management: Wallboard maintains a robust certificate management process. TLS certificates are issued by trusted certificate authorities and renewed before expiration. Certificate validation is strictly enforced to prevent man-in-the-middle attacks.

Avoiding Homegrown Cryptography: Wallboard steers clear of using custom or homegrown cryptographic algorithms. Instead, we rely on well-established libraries and frameworks that have undergone extensive security testing and peer review.

Regular Cryptographic Reviews: Our development team conducts regular cryptographic reviews to identify and mitigate potential weaknesses in the encryption and decryption processes. This proactive approach helps maintain a strong security posture.

Handling Cryptographic Errors Securely: Wallboard implements secure error handling in cryptographic operations to avoid potential information leakage that could aid attackers in exploiting vulnerabilities.

Data validation

Parameterized Queries: Wallboard utilizes parameterized queries and prepared statements in database interactions. By separating data from the query logic, we prevent malicious input from being treated as code, mitigating the risk of SQL injection attacks. The requirements are achieved by using modern frameworks like Spring, Hibernate and JooQ.

Input Validation and Sanitization: All user-supplied input is thoroughly validated and sanitized before being processed. We enforce strict input validation rules to ensure that only expected and safe data is accepted by the application.

Escaping Output: Data rendered in HTML or other output formats is properly escaped to prevent cross-site scripting (XSS) attacks. This ensures that user-supplied data is not mistakenly interpreted as code when displayed to other users.

Secure API Design: Wallboard follows secure API design practices. API inputs are validated and sanitized before processing, and access to sensitive APIs is appropriately restricted through proper authentication and authorization mechanisms.

Security Patch Management: Regularly applying security patches to the underlying technologies used by Wallboard is crucial. This includes web servers, frameworks, databases, and other components to address known vulnerabilities that could be exploited for injection attacks.

Design and development

Threat Modeling: During the initial phases of development, Wallboard conducts a comprehensive threat modeling exercise. This involves identifying potential security threats and vulnerabilities that may arise due to the application's design. By proactively identifying these risks, we can design countermeasures to mitigate them effectively.

Secure Architecture Patterns: Wallboard employs secure architecture patterns and follows

industry best practices for cloud-based applications. We emphasize modular and layered designs that segregate sensitive components from public-facing ones, reducing the attack surface.

Security by Design: Security is an integral part of the development process at Wallboard. Our design principles prioritize security considerations and ensure that security controls are embedded within the application's architecture and functionality.

Secure Integration with Third-Party Services: When integrating with third-party services, Wallboard performs due diligence to ensure the security and trustworthiness of these services. We validate their security practices and adhere to secure integration guidelines.

Configuration

Secure Configuration Management: All system configurations are carefully managed and maintained. Security configurations are defined and consistently applied to prevent misconfigurations that could lead to security vulnerabilities.

Secure Defaults: Wallboard employs secure default configurations for all components and frameworks used in the application. By starting with secure defaults, we reduce the chances of unintentionally leaving sensitive features exposed.

Centralized Configuration Management: Configuration settings are managed centrally, minimizing the risk of inconsistencies and ensuring that security settings are uniformly applied across all environments.

Secure Cloud Platform Configurations: If Wallboard is hosted on a cloud platform, we adhere to the cloud provider's best practices for securing configurations. Cloud resources are configured securely to prevent unauthorized access and data exposure.

Logging and monitoring

Error Handling and Logging: The application incorporates a sophisticated error handling system that prevents sensitive information from being exposed in error messages. Extensive logging and monitoring capabilities enable the identification of potential access control issues and swift response to suspicious activities.

Comprehensive Logging Policy: Wallboard maintains a comprehensive logging policy that defines what events are logged, where they are stored, and how long they are retained. This policy ensures that relevant security events are consistently captured for analysis.

Log Integrity and Protection: Wallboard ensures the integrity and protection of logs. Access to log files is restricted to authorized personnel only.

Real-time Monitoring: Security events are monitored in real-time, allowing for immediate detection of suspicious activities or potential security breaches. Automated alerts and notifications are set up to inform the appropriate personnel promptly.

Regular Log Analysis: Security logs are regularly analyzed for suspicious patterns or anomalies. This proactive approach helps detect potential security incidents before they escalate.

Continuous Monitoring of Critical Systems: Critical systems and infrastructure are continuously monitored to detect any security-related issues or unauthorized access attempts in real-time.

Periodic Security Assessments: Regular security assessments and penetration testing are conducted to validate the effectiveness of the logging and monitoring system. This helps identify any gaps or weaknesses in the security logging process.

Server host security

When it comes to Linux server security, general recommendations are always applied on the server side. Wallboard minimizes the number of running applications, removes any unneeded applications, and closes all unnecessary ports through the firewall to prevent security breaches. Despite Wallboard's application being designed to be secure and robust, it is advised using an additional separate cloud firewall appliance in front of the server to enhance security measures.

Wallboard strongly recommends using a dedicated server with no other applications as an additional security measure. By opting for a dedicated server environment, not only does Wallboard enhance security, but it also prevents potential conflicts with other server applications. This dedicated setup ensures that the server resources remain exclusively allocated to Wallboard's applications, minimizing the risk of interference or compatibility issues that could arise when sharing resources with other applications. This approach guarantees optimal performance, stability, and isolation, providing an added layer of protection for Wallboard's server environment.

In terms of security updates, Wallboard emphasizes the importance of automatically installing them, following the Security Announcements specific to the installed operating system. By doing so, the server remains up to date with the latest patches and effectively addresses any identified security vulnerabilities.

Remote access

Wallboard uses SSH to establish remote connections to the server for maintenance tasks. The scope of permitted IP addresses is restricted solely to Wallboard's office IP address. Access to SSH keys is granted to specific employees, and each key is fortified with a passphrase randomly generated for enhanced protection. SSH keys and their corresponding passphrases are stored separately. Wallboard ensures the secure storage of passwords by utilizing a dedicated password manager server. This specialized server is designed to securely store and manage passwords, employing encryption techniques and access controls.

It is worth noting that SSH keys utilized by Wallboard are at least either 2048 or 4096 bits in length, ensuring modern security measures.

Docker

Wallboard's server operates within a Docker environment, leveraging the versatility and efficiency it offers. By utilizing containers, Wallboard can encapsulate the entire application along with its necessary components, including application servers, microservices, databases and other dependencies. This allows them to bundle everything into a single package, simplifying the deployment process. Consequently, Wallboard can efficiently ship the entire application ecosystem as a cohesive unit, ensuring consistent functionality and seamless installation.

Docker and the local firewall

In a Linux environment, Docker utilizes iptables rules to achieve network isolation, effectively enhancing security measures. Wallboard recommends against modifying the rules that Docker inserts into your iptables policies.

It is important to note that if the server employs a local firewall, such as UFW or Firewall, Docker will override it. When Docker publishes a specific port, such as port 443 for TCP traffic, it takes precedence over the firewall configuration settings that were previously established. This means that Docker's port publication supersedes any firewall rules associated with that particular port.

Understanding this behavior is essential for system administrators, as it ensures they are aware that Docker's port publication takes precedence over the local firewall settings. By acknowledging this, they can ensure the network configurations align with their intended setup while maintaining a clear understanding of how Docker interacts with the local firewall.

Server infrastructure

Wallboard uses Terraform as Infrastructure as Code (IaC) tool to facilitate the installation of the server components and implement container updates. By utilizing Terraform, Wallboard can automate the deployment and management of their infrastructure in a consistent and efficient manner.

Terraform scripts and modules utilized by Wallboard are securely stored within a private Git repository. This private repository ensures that access to the infrastructure code is restricted to authorized individuals within the Wallboard organization.

Opened ports

The server has specific ports opened to facilitate various functionalities. These ports are mandatory for users to log in to the Management Interface and for player applications to successfully play contents.

Among these ports, port 22/TCP is exclusively utilized for maintenance purposes. To maintain a higher level of security, access to this port is restricted to Wallboard's office IP address.

Web application	80, 443/TCP - inbound
NTP time server	123/UDP - inbound
OpenSSH	22/TCP – inbound, restricted to Wallboard's office IP address

System recommendation

For detailed information regarding the recommended system specifications, please visit our password protected documentation at https://private.docs.wallboard.us/docs/sys-admin/server-installation/prerequisites/.

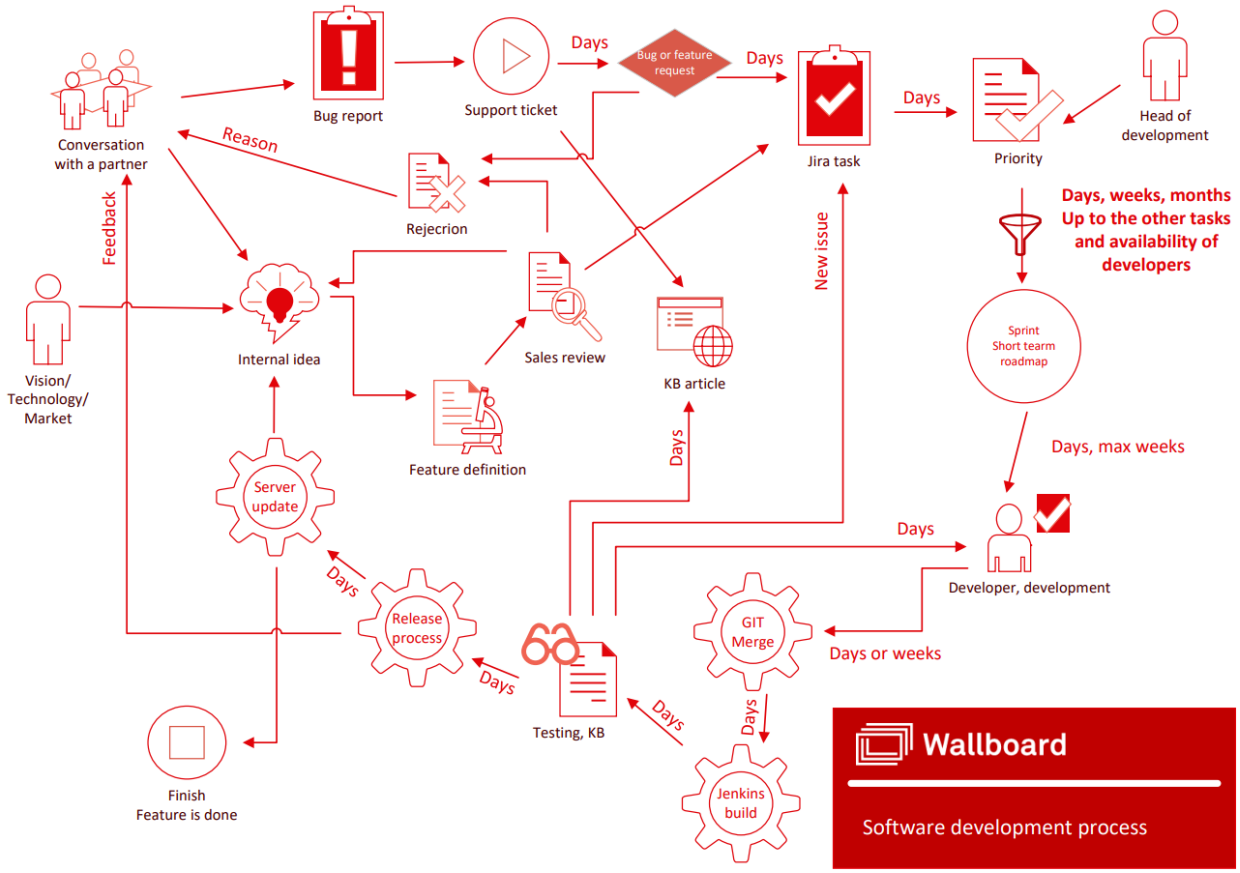
Application Development Security – Methods

Secure coding practices

Wallboard is committed to following OWASP Secure Coding Practices, ensuring that the code adheres to industry-leading standards for security. Furthermore, as part of the code integration process, Wallboard requires that the code passes the SonarQube security analysis with the highest level of achievement possible. This approach guarantees that Wallboard's code undergoes rigorous scrutiny and analysis to identify and rectify any potential security vulnerabilities. By combining adherence to OWASP Secure Coding Practices with the stringent requirements of SonarQube, Wallboard places a strong emphasis on creating secure and robust software solutions.

Process

Wallboard places emphasis on not only ensuring the security of their applications but also prioritizing secure development processes. This holistic approach entails integrating security measures and best practices into every stage of the development lifecycle. By focusing on secure development processes, Wallboard aims to proactively identify and mitigate security risks from the initial stages of software development, leading to more resilient and trustworthy applications. This comprehensive approach aligns with industry standards and helps establish a culture of security throughout Wallboard's development practices.



Application Security - Server

Web server application

The server application comprises multiple components, such as the main server app and various microservices. To make sure these services are not directly exposed, a reverse proxy is implemented in front of the application. This reverse proxy efficiently handles incoming requests, including the ability to proxy requests and establish WebSocket tunnels to the relevant internal services. By default, all requests are automatically redirected to HTTPS, ensuring secure and encrypted communication between the client and the server. This setup enhances the overall performance, security, and reliability of the server application.

For security reasons, Wallboard exclusively utilizes secure TLS 1.2 and 1.3 protocols while purposefully excluding weak ciphers. Wallboard ensures that all communication channels between their server application and clients are protected by strong encryption standards. By excluding weak ciphers, Wallboard mitigates potential vulnerabilities and reinforces their application security measures, providing a safer environment for users and safeguarding sensitive information.

Databases

Wallboard utilizes MySQL as the primary database to store customer data, system and customer configurations. To effectively capture and store device metrics, InfluxDB is used. Lastly, Wallboard utilizes TimescaleDB to securely store proof of play statistics. This strategic use of different databases enables Wallboard to efficiently manage and retrieve specific data types, ensuring optimal performance and organization of their diverse data sets.

In order to mitigate the risk of potential data breaches, Wallboard ensures that none of the databases are exposed externally. By maintaining strict access controls and network configurations, Wallboard significantly reduces the likelihood of unauthorized access or potential security vulnerabilities, reinforcing the overall security of their data storage infrastructure.

Customizations

PM2 is a Node.js process manager that Wallboard utilizes to effectively manage and serve custom widgets. In order to access PM2 admin functions, such as uploading a new module, it is required to have a Wallboard ADMIN account. The authorization server implemented by Wallboard verifies the token to ensure proper authentication and authorization for accessing PM2 admin functionalities.

However, it's important to note that the security of any other module uploaded through PM2 is the responsibility of the customer. While Wallboard provides the necessary authentication mechanisms and safeguards through the authorization server, the customer assumes the responsibility of ensuring the security and integrity of the modules they upload.

N8N is a workflow automation tool that is **no longer deployed by default** in Wallboard's system. However, for those who choose to utilize it, the administration interface of N8N is protected by basic authentication. This means that users are required to provide valid credentials to access and manage workflows through the administration interface. This additional layer of authentication helps ensure that only authorized individuals can access and make modifications to the

workflows within N8N.

Nodemon/Node.js have been the **legacy approach** for deploying customization modules on Wallboard. Wallboard has transitioned to using PM2 as a replacement. PM2 offers advanced features and improved functionality for deploying and managing customization modules in the current setup.

Full environment

Wallboard maintains a robust server environment consisting of various components. They prioritize the regular application of security updates to ensure the overall security and integrity of their system.

- Java based web application
- Java based native micro services
- MySQL – main database
- InfluxDB – device metrics database
- TimescaleDB – proof of play database
- nginx – reverse proxy
- certbot – automatic issuance and application of SSL/TLS certificates
- PM2 – NodeJs process manager to manage and server custom widgets and components
- N8N and database – legacy workflow automation tool
- chrony – NTP server to provide server time to devices
- autoheal – restarts unhealthy containers
- Prometheus – store server/application metrics
- Exporters – provide server/application metrics
- Grafana – monitoring, alerting, visualize Prometheus data

Optional components

- rsnapshot – create application-level server backups
- Elasticsearch database – store logs
- Filebeat – read and collect container logs

Remote monitoring and alerts

To ensure the smooth operation of Wallboard servers and promptly detect any issues, Wallboard relies on the third-party monitoring tool Grafana. This powerful tool enables comprehensive monitoring of various metrics essential for server performance. The following metrics are primarily collected and monitored:

- CPU usage: Tracking the utilization of the server's processing power to identify any spikes or abnormal behavior.
- Memory usage: Monitoring both free memory and swap statistics to ensure efficient memory management and avoid potential bottlenecks.
- Network usage: Keeping a close eye on network activity to identify any unexpected fluctuations or anomalies.
- Disk usage: Monitoring the utilization of disk space, free inodes, and disk I/O performance to prevent storage-related issues.

- Web server application metrics: Gathering relevant data on the performance and behavior of the web server application to optimize its functionality.
- Database statistics: Capturing metrics such as connected clients, performance data, slow queries, and index usage to maintain the database's efficiency and address any potential bottlenecks.

By actively monitoring these metrics through Grafana, Wallboard can proactively identify any deviations or anomalies that may indicate server issues, enabling them to take prompt action and ensure the uninterrupted operation of their system.

Licensing policy and security

A valid license is required for the Wallboard server to start operating. This license is signed using a private key, which is securely stored solely on the license server managed by Wallboard. The server verifies the license's content by communicating with Wallboard's license server.

During the startup process, the server performs a thorough license check to ensure its validity. Additionally, subsequent checks are conducted every hour to maintain the licensing integrity. To uphold the highest security standards, all communication between the server and the license server occurs over a secure HTTPS connection.

By implementing these measures, Wallboard guarantees that the server operates with authorized licenses, promoting accountability and preventing unauthorized usage. The secure communication protocol ensures the confidentiality and integrity of the license verification process, further bolstering the overall security of the Wallboard server.

Edge Server

Wallboard Edge Server is a standalone server component, that is based on Apache Traffic Server CDN. Edge Servers play a crucial role in optimizing content distribution and scaling, particularly in low-bandwidth internet locations. By strategically positioning these servers, Wallboard effectively minimizes network latency and enhances content delivery performance. The Edge Servers intelligently handle content distribution, optimizing the delivery process to provide a seamless and efficient experience even in challenging network environments.

Server Information Security

To maintain data security, the server diligently stores all sensitive information in databases, while uploaded files are kept in its designated working directory on the file system. Wallboard's SaaS infrastructure, hosted on Digital Ocean, takes data protection a step further by storing all data on encrypted block volumes. This additional layer of encryption enhances the confidentiality and integrity of the stored information.

When it comes to uploaded media files and datasources, Wallboard employs a unique file identifier that prevents unauthorized guessing of file locations. Media files and datasources are stored and served based on these unique identifiers, ensuring controlled access. The original file names remain stored exclusively in the database, maintaining an added level of confidentiality and preventing unauthorized disclosure.

As a best practice, Wallboard consistently advises all partners utilize Wallboard’s backup solution to perform regular backups and store them securely on an independent network drive. This precautionary measure helps protect against potential data loss and provides a reliable restore point in the event of unforeseen circumstances or system failures.

Additionally, Wallboard strongly recommends that partners inform their end customers about the importance of exporting their contents from the system and independently archiving them. By doing so, end customers can maintain their own copies of valuable content, safeguarding against any potential data loss or system disruptions. This approach empowers end customers with control over their own data and ensures that they have the ability to restore their content independently, further enhancing the overall data protection strategy.

Application Security - Management User Interface

For secure authentication and authorization, the authorization protocol employed between the server and the front-end Management User Interface (UI) adheres to the widely recognized OAuth2 standard. By utilizing OAuth2, Wallboard ensures a standardized authentication mechanism, providing secure access to the Management UI while protecting sensitive user data. This industry-standard authorization protocol strengthens security and enhances the overall system integrity.

Application Layer Security

In order to access the Management User Interface (frontend), users must log in to the system using their unique username and password. Once authenticated, the frontend establishes secure communication with the REST API (backend) through authenticated requests.

During the authorization flow, several checks are conducted to ensure the validity and appropriateness of the user's token. These checks include verifying that the token has not expired, confirming that the user or their client is neither disabled nor expired, and validating that the user possesses the necessary privileges to perform the intended operation.

To ensure appropriate user capabilities and access privileges, the Wallboard system offers a range of predefined user roles. These roles restrict users to specific functionalities and determine their level of authority within the system. The available user roles include:

- Administrator
- Owner
- Technician
- Approver
- Editor
- Viewer

For the complete list of capabilities, please visit the following Knowledge Base article: https://www.wallboard.us/knowledge-base/user-roles.

To further organize and define user access, Wallboard utilizes Teams that effectively separate users based on their access requirements. This team-based approach ensures proper access management and enhances security within the Wallboard environment.

The management interface of Wallboard is specifically designed to be compatible with modern browsers, ensuring a seamless user experience and optimal functionality. To enhance security, HSTS (HTTP Strict Transport Security) policy is implemented, which enforces the use of secure and encrypted connections, reducing the risk of unauthorized access and data interception.

To mitigate potential security risks, custom scripts are carefully executed only in unauthorized browsing sessions.

Password Policy

To maximize security, Wallboard employs encryption during password transmission and hashing when passwords are stored. These measures protect passwords from unauthorized access and add an extra level of security.

It is crucial to note that Wallboard employees do not have access to user passwords, maintaining the privacy and confidentiality of user credentials. This reinforces the commitment to user security and ensures that passwords remain inaccessible to anyone within the organization.

Wallboard follows a default password policy that promotes strong password practices. This policy requires passwords to be a minimum of 8 characters in length and include at least one uppercase letter, one lowercase letter, and one number. These requirements contribute to the creation of resilient passwords.

Furthermore, Wallboard administrators possess the flexibility to customize and adjust the password policy according to their specific security needs. This adaptability allows administrators to align the password requirements with the unique requirements and risk profile of their organization.

Brute force detection and protection

To enhance security and mitigate brute force login attacks, Wallboard servers enforce restrictions on the number of failed login attempts permitted within a specified timeframe. These limitations can be customized by Wallboard administrators, granting them the flexibility to configure appropriate security measures that align with their needs.

The server can restrict requests based on IP addresses, ensuring that suspicious or repeated login attempts from specific addresses are effectively managed. Additionally, it can also limit requests for a specific user, further fortifying the authentication process and preventing unauthorized access.

Two-Factor Authentication

Wallboard users have the option to enable two-factor authentication (2FA) for enhanced security. This feature utilizes a time-based one-time password (TOTP) mechanism, which can be generated using any compatible Authenticator application. By implementing 2FA, Wallboard adds an additional layer of protection to user accounts, requiring the generation and verification of a unique TOTP in addition to the regular username and password. This helps prevent unauthorized access and reinforces the overall security posture of the system.

Single Sign-on

Wallboard provides Single Sign-On (SSO) using Wallboard's authentication server (KeyCloak). The communication between the authentication server and Wallboard is consistently encrypted to ensure secure transmission of data. As part of the SSO process, callback URLs need to be whitelisted, adding an additional layer of security to verify and authorize the appropriate authentication flow. This comprehensive approach to SSO guarantees the protection of user credentials and enhances the overall security of the system.

Application Security – Client applications

The player applications authenticate themselves on the server using an immutable 32-length hex string. This unique identifier serves as a secure authentication mechanism, ensuring that only authorized player applications can access and interact with the server. Additional security measures such as token authorization are currently not implemented in the system. Wallboard continuously evaluates and explores options to enhance security measures, ensuring that the system remains secure and resilient to potential threats.

Device locking refers to the process of modifying system resources and settings to prevent users from exiting the application unintentionally. This security measure ensures that the device remains dedicated to displaying the intended content and prevents unauthorized access to other device functions or applications.

Android Client

The Wallboard Android client is designed to be compatible with most Android-based devices running on Android 5.0 (API Level 21) or above, ensuring widespread accessibility across a range of devices.

To ensure the best experience and compatibility, it is advisable to consult the Knowledge Base for a list of recommended device manufacturers and models. This resource provides valuable information and insights on specific device models that have been tested and proven to work well with the Wallboard Android client.

On Android, device locking can be achieved either through the manufacturer's provision or by manual configuration. Supported devices are locked by default, thanks to the efforts of the manufacturers. This default locking configuration ensures that the devices are ready to be securely and efficiently used with the Wallboard application from the moment they are powered on.

When opting for manual device locking, Wallboard offers a solution that requires a rooted device. In this case, Wallboard removes the default Home application and the System UI, ensuring that the device remains focused on running the Wallboard application exclusively.

Brightsign Client

BrightSign clients, as part of a dedicated signage platform, are inherently locked down by nature. They lack a user interface and do not run any other applications on the system, which further enhances their security. By eliminating unnecessary software and interfaces, BrightSign clients minimize the attack surface and potential security vulnerabilities.

LG Client

The LG WebOS client is specifically designed for LG signage TVs, which are inherently locked by nature as they prevent the execution of other applications in the foreground and limit access to the system. This locked-down environment ensures the secure and reliable operation of the LG WebOS client within the LG signage ecosystem.

As part of the device lockdown process, the LG WebOS client includes the ability to turn off the remote controller (IR). This additional security measure further enhances the overall system

security by disabling external control and preventing unauthorized access or interference.

Samsung Client

The Samsung Tizen client is exclusively tailored for Samsung signage TVs, which inherently employ a locked-down approach, preventing the execution of foreground applications and limiting system access. This inherent locking mechanism ensures a secure and dependable operation of the Samsung Tizen client within the Samsung signage ecosystem.

As part of the device lockdown process, the Samsung Tizen client includes the ability to turn off the remote controller (IR). This additional security measure further enhances the overall system security by disabling external control and preventing unauthorized access or interference.

Windows Client

The Windows application is specifically designed to be compatible with the Windows 10/11 operating systems, ensuring optimal performance and functionality on these platforms.

To lock down the device, there are two available methods. The first is using the Windows Shell Launcher, which provides a convenient and straightforward way to restrict access and prevent users from exiting the application. The second method involves manual system modifications, allowing for more advanced customization and control over the device lockdown settings.

Once the device is locked down, various measures are implemented to ensure the application remains uninterrupted. This includes disabling the use of edge gestures, preventing users from easily exiting the application. Additionally, key combinations that typically allow for application termination are disabled, further enhancing the device lockdown.

Desktop Broadcast App

The Desktop Broadcast App, unlike traditional signage applications for Windows 10/11 PCs, does not necessitate the locking of the device or the prevention of users from utilizing other functionalities on their machines. As a result, the application does not require elevated privileges to run and can operate within the normal user context.

During the installation process, elevated privileges are required to allow the installer to install the application to the *Program Files* folder with a *perMachine* scope. Custom actions are employed by the installer to initiate the application after installation or to close any currently running instances of the application during the installation or update process. These actions are executed within the *user context* and can be bypassed using command line arguments if desired.

Additionally, the application provides the option to scan and report foreground applications to the server, although this reporting module is not installed by default. To enable this feature, specific command line arguments can be passed to the installer.

Application data, including settings, logs, and cache files, are stored in the user's *Roaming* folder. This ensures that user-specific data is preserved and accessible across different devices and sessions.

Sensors in general

The sensor application is a firmware specifically designed to operate on M5/ESP32 devices, providing sensor functionality for the system. To communicate with the client application, the sensor application utilizes a USB cable, establishing a connection through a specified JSON-based protocol.

For optimal performance, each sensor application requires its own specific configuration that outlines the sensors connected to it. This configuration, detailing the sensor setup, can be conveniently managed and set through either the Sensor Designer Application or the Management User Interface provided by Wallboard.

Physical Security

Wallboard always advises taking measures to restrict physical access to the devices, as it plays a crucial role in enhancing overall security. It is recommended to hide the cabling and power plug devices, if feasible, to prevent tampering or unauthorized interference. Additionally, ensuring that USB connectors are not accessible to unauthorized individuals is important to maintain the integrity of the system.

To further bolster security, it is advisable not to leave keyboards or pointing devices connected to the player device when they are not required. This reduces the potential risk of unauthorized access or input.

Summary

As the sole developers of the server software and client app components, Wallboard is dedicated to upholding and surpassing current security best practices to enhance the overall security of our system. We are committed to continually improving our system based on evolving best practices and valuable feedback received from our partners.

By adhering to rigorous security standards, we ensure that our system remains resilient against emerging threats and vulnerabilities. We proactively monitor industry developments and incorporate the latest security enhancements into our software and applications. This ongoing commitment allows us to provide a secure and robust digital signage solution for our partners.

We highly value the feedback and input from our partners, as it serves as a vital source of information for identifying areas of improvement. By actively engaging with our partners and considering their valuable insights, we can make informed decisions and implement necessary security enhancements to further fortify our system.

Through our dedication to continuous improvement and collaboration with partners, we strive to deliver a highly secure digital signage solution that instills confidence in our partners and ensures the protection of their valuable assets.