



Product Security

Contents

MegaMeeting v.4 WebRTC.....	3
MegaMeeting v.2 and v.3 Adobe Flash	3

Product Security

MegaMeeting v.4 WebRTC

All MegaMeeting v.4 Meetings are browser-based and center around WebRTC for audio and video streaming.

1. End-to-End Encryption

Encryption is built in to WebRTC and is mandatory for all components including signaling. Peer-to-peer transmission is encrypted using standards supported by compatible web browsers.

2. Data Transport Layer Security (DTLS)

Encryption is handled using DTLS which is based on the TLS standard and is designed to prevent eavesdropping, tampering, and message forgery.

3. Secure Real-Time Protocol (SRTP)

Additional encryption is handled by SRTP which protects real-time streaming data (such as video and audio) and is intended to provide encryption, message authentication and integrity, and replay attack protection.

4. Camera And Microphone Access

Access to camera and microphone devices is handled directly by the user's web browser and explicit permission must be granted by the user before camera or microphone data can be accessed by the web browser. The user remains in control of camera and microphone access at all times, and can disable access to camera and microphone devices using the controls provided by their web browser.

5. WebSockets (ref)

WebRTC provides secure signaling channels for voice and data communication over web sockets.

6. Meeting Security

All meetings can be additionally secured via unique access key per participant, which prevents unauthorized access to a meeting. Keys can be revoked at any time to prevent future access to a meeting.

7. Account Security

All accounts are accessed over secure HTTPS (SSL) connection and protected by authentication. User credentials are stored as cryptographic hashes.

MegaMeeting v.2 and v.3 Adobe Flash

All MegaMeeting v.2 and v.3 Meetings are browser-based and center around Adobe Flash Player.

1. Flash Player Security and HIPAA Compliance

Product Security

In a world where most digital experiences fall flat, the Adobe Flash technology offers something different. It's a lightweight, cross-platform runtime that can be used not just for rich media, but also for enterprise applications, communications, and mobile applications. The Flash technology is fueling an increasing number of Rich Internet Applications (RIAs). And as a result a growing number of employees, partners, and customers have access to enterprise data and processes. This access, combined with the requirement to comply with industry regulations such as the Sarbanes-Oxley Act and the Health Insurance Portability and Accountability Act (HIPAA), has enterprises interested in the level of security provided by this framework. The Flash technology and the Flex product family address this concern by leveraging an organization's existing security solutions and technologies.

The Adobe approach is to implement robust security within its own products while avoiding new exposures to the rest of the environment. However, the Flash technologies are not security products—they leverage existing security tools and approaches that are already in place, while minimizing additional investments in security. Flash was designed to be inherently secure, leveraging industry standard security procedures to deliver a reliable user experience. For example, the Flash technology integrates seamlessly into an organization's existing architecture at the browser level through a plug-in and at the presentation tier through Flex software or a static HTML solution with script and Flash.

The Adobe Flash technology leverages an organization's existing infrastructure. Security is handled by existing security solutions and protocols. Because the Flash technology leverages SSL and authentication technologies and requires no changes to access control or other security settings, organizations do not need to deploy additional security solutions to use the Flash technology. In Flash environments, security is handled by existing security solutions and protocols.

The Flash technology is a true multiplatform environment that leverages the core security capabilities of the underlying operating systems, browsers, and application servers. The Flash technology is based on proven and accepted security standards such as SSL and HTTPS for data transport. It has a layered architecture that encompasses these key elements. This paper focuses on the servers and runtimes (for example, Adobe Flash Player and Adobe Flex software), which are used to deliver Flash applications, content, and communications, and which act as the platform, provide the controls, and specify the architecture.

Due to the increasing pressures to comply with a range of industry regulations and the fact that a growing number of partners, contractors, and customers have access to corporate networks, enterprises are investing significant amounts in authentication and authorization services. These include single sign-on, VPN integration, specialized hardware (for example, smart cards), PKI, RSA, SecurID®, or other physical tokens. At the same time, industry-specific requirements are mandating organizations to deploy authentication solutions. For example, both federal agencies and financial services organizations are required to utilize two factor authentication measures to secure electronic transactions. Similarly, pharmaceuticals and health care organizations are facing tremendous pressure to protect the privacy of individuals through regulations such as HIPAA. Fortunately, organizations that use the Flash technology can leverage their existing infrastructure and security investments to address these requirements. Flex Data Services sits on top of a Java server and integrates with standard protocols for authentication, such as LDAP and other directory services. On the client side, the Flash client runtime takes advantage of the common security technologies available in web technologies, such as the trans-

Product Security

parent authentication handling by browsers.

In addition to authentication, access control is increasingly being used to determine who has access to which content and applications within a corporate network. While access control requirements vary by application, the Flash technology incorporates a number of features that help organizations address these needs. Some of these access control features come pre-set, and in some cases, administrators or users can customize them to their needs. Server-Side Access Controls through the Flex Data Services, the Flash technology offers access control to server-side data by utilizing existing access controls on the host servers. In addition, administrators can control access to all data service destinations. You can protect HTTP- and RTMP- based endpoints by using firewall/router/webserver IP whitelists and blacklists. A whitelist contains client IP addresses that are permitted to access endpoints. A blacklist contains client IP addresses that are restricted from accessing endpoints. The blacklist takes precedence over the whitelist in the event that the client IP address is a member of both the whitelist and blacklist.

Client- Side Access Controls, much like the model employed for Java and JavaScript, Flash Player runs content inside a virtual machine that implements a security sandbox. Within this sandbox, all Flash Player resources (applications, data, network URLs, and so on) are essentially isolated from the rest of the computing environment, as well as other sandbox instances. This approach provides an advantage over traditional web-enabled applications, such as ActiveX solutions, which often have complete access to the operating system environment. While Flash Player applications may interact freely with resources within the same sandbox, the Flash Player sandbox prevents unauthorized access to the operating system environment as well as other local instances of Flash Player.

2. Unauthorized Access to Data

Unauthorized access to data refers to data on local disks, networked disks, or web servers that are communicated over the network or stored in memory by an application or process (for example, password lists, address books, privileged documents, and application code). An ActionScript program in Flash Player cannot write, modify, or delete any files on the client machine other than shared objects (small, Flash-specific files), and it can only access shared objects on a per-domain basis. Internet-based Flash applications cannot read any other local files, or any sensitive or private data. In fact, no ActionScript methods available to Flash applications can create, modify, or delete directories or files directly. In order for web-based Flash Player content to access server data, the domain serving the Flash Player content must get explicit permission from the domain hosting the requested data (AKA the provider domain). Without permission, the load will fail. These permissions are specified by a policy file located on the server of the provider domain. This file enables access control by explicitly listing the domains that have permission to access data on that server.

3. Unauthorized Access to Private User Information

Personal and financial data – as well as information about the user’s security settings for Flash Player – often resides on a user’s machine, and users are rightly concerned about others accessing this information. However, users should be aware that Flash Player does not collect information about them. Users have control over the Flash Player behavior when encountering decisions concerning privacy. Through the Flash Player Settings user interface and Settings Manager, users can fine-tune the following settings related to privacy and security:

Product Security

Local storage of data using the local shared objects mechanism
Access to cameras and microphones connected to the system
Notification of updates to Flash Player

In an enterprise environment, network administrators can control settings for Flash Player centrally to ensure that all clients conform to the corporate security policy. In addition to the fundamental protections provided by the sandbox and virtual machine, the Flash Player client also provides stakeholders (those who own or administer a resource) with flexible, easy-to-use controls to permit (or limit) access to sensitive resources such as network files and databases. The Flash Player security model is organized in a way that enables enterprises to delegate control of permissions to the appropriate stakeholder. This model also supports the distributed architectures that are commonly used for applications built on the Flash technology.

4. Malicious Code

All organizations face the potential for malicious code infection that can spread quickly throughout the corporate network. For example, Internet users could download what appears to be a legitimate program that in reality carries a threat such as a Trojan Horse program, which could expose the network to hackers. Or code authorizing remote access to a network can reside unnoticed in browser cookies or Web applets. Adobe Flash Security and Adobe Enterprise Solutions utilize a “Sandbox Approach” which allows for protection against malicious code and activity. As discussed previously, because of the sandbox security approach on the client side and the use of Java on the server side, the Flash technology uses in-place security tools to maintain resistance to malicious code, such as viruses, Trojan Horse programs, back door worms, and spyware. In addition, the design of Flash Player includes architectural characteristics that minimize malicious code threats compared to ActiveX or JavaScript solutions. Because all Flash Player resources are isolated from the rest of the computing environment — as well as other sandbox instances — through the sandbox approach, the host system is protected against malicious activity and potentially harmful programs and content. In fact, in a memorandum from the Joint Chiefs of Staff regarding policy guidance for the use of mobile code technologies in the Department of Defense (DoD) information systems, Flash Player is listed under Category 3, the most secure of the three categories.

5. Minimized SQL Injection and Cross-Scripting Vulnerabilities

Solutions that use runtime interpreted string-based languages — such as JavaScript and DHTML — are especially susceptible to SQL injection and cross-site scripting, which both are listed among the top 10 vulnerabilities on the Open Web Application Security Project site (Source: www.owasp.org). In contrast, Flash content is delivered as a series of instructions in binary format to Flash Player over web protocols in the SWF file format. The SWF files themselves are typically hosted on a server and then downloaded to, and displayed on, the client computer when requested. Because Flash Player is binary and compiled, it inherently minimizes these threats compared to string-based language solutions that may leave back-end data vulnerable and unprotected. Typically, applications access databases through dynamically generated SQL statements, because these statements are fairly easy to implement and provide for looser coordination with the database. However, it is difficult to produce dynamically generated SQL statements that are resistant to SQL injection. In addition, dynamic statements often re-

Product Security

quire broad access permissions to database objects. Prepared statements protect against SQL injection, while stored procedures allow the database to be more tightly locked down. During the application penetration assessment conducted by Symantec Professional Services mentioned previously, Symantec found that the implementation of stored procedures prevented attempts to compromise application data through the use of SQL injection and manipulation attacks.

6. Data Transport

Clearly, the secure transport of data between Flash and Flex hosts and applications is critical to ensuring the integrity of the data, as well as making sure others do not use that data for malicious purposes.

7. Standards Compliance

Both Flash Player and the Flex product line use standards-based protocols for data transport. Flash Player knows whether its data was obtained over a secure HTTPS (HTTP over Secure Sockets Layer) connection and records that fact using separate sandboxes. Data loaded from HTTPS sites is subsequently treated differently than data from HTTP or other, less secure sources. This client data segmentation is a natural extension of the most common PKI models, which use x509 certificates to identify clients and servers. Cryptographic standards such as x509 certificates are implemented by the browsers with which Flash Player interoperates. On the server side, these standards are implemented by the hosting environment. By using XML and SOAP standards for data transport, the Flex product line benefits from common security technologies such as HTTPS, which is supported for all operations.

8. Wireless Security

As the corporate network extends to provide access to a variety of constituents — such as contractors, partners, customers, and telecommuters — organizations must protect an increasing number of remote users. Without effective wireless security, not only is the data in transit vulnerable to access and manipulation, but the enterprise network itself is vulnerable to Internet threats and malicious code that can be introduced through wireless devices. By using SSL, native encryption, and the security on the operating system, Flash Player and the Flex product line minimize wireless security concerns.

Since Flash applications running within a browser use the browser for almost all communication with the server, they can take advantage of the browser's built-in SSL support for encryption. In addition, the actual bytes of a Adobe Flash application can be encrypted while they are being loaded into the browser. By playing a Flash application within an SSL-enabled browser through an HTTPS connection with the server, organizations and users can ensure that the communication between Flash Player and the server is encrypted and secure.

9. Ease of Integration with SSL Accelerators and Load Balancers

Integration with SSL accelerators and standard load balancers is simple. For example, because Flex Data Services handles requests that are initially received by a web server, the Flex server does not need to know what protocol is being used. To switch from HTTP to HTTPS, the server administrator simply modifies the web server as he or she would have done without the Flex server installed.

Product Security

10. Support for Encrypted Tunneling

Applications built with Flash Media Server use the Real-time Messaging Protocol (RTMP) for high performance transmission of audio, video, and data messages in a single data channel between the client and the server. While RTMP does not include security-specific features, Flash communications applications can perform secure transactions and secure authentication through an SSL-enabled web server. When running within a browser, Flash Player can use secure encrypted HTTPS tunneling to communicate through RTMP. This tunneling support provides users behind a typical corporate firewall with a transparent experience while ensuring secure data transport.

11. Conclusions

With the Flash technology, organizations can develop, deploy, and distribute with confidence RIAs, enterprise and mobile applications, and communications to employees, partners, and customers. Flash Player and the Flex product line leverage an organization's existing security infrastructure (which means they are security independent) are based on existing accepted standards, and use secure technologies. By virtue of the way that the Flash technology and the Flex product line integrate with existing authentication, access control, data transport, and malicious code prevention solutions, they do not adversely affect an organization's ability to meet security requirements. Just as importantly, this approach supports continued compliance security best practices and regulations, such as the Sarbanes-Oxley Act of 2002 and HIPAA. And by leveraging an organization's existing security infrastructure, the Flash technology enables the successful deployment of secure applications without further investments. According to an independent security assessment by @stake, Adobe has developed a strong information protection model against client-side threats. "[The Flex] architecture mitigates many common client-side attacks such as cross-site scripting, denial-ofservice [attacks], SQL injection, man-in-the-middle [attacks], and session hijacking." In addition, server-side security is maintained by leveraging J2EE security to mitigate common attacks against infrastructure components, such as buffer overflows, heap corruption, and cross-site scripting.

For More Information, please visit
<http://www.adobe.com>

Adobe, the Adobe logo, Acrobat, Clearly Adobe Imaging, the Clearly Adobe Imaging logo, Illustrator, ImageReady, Photoshop, and Post-Script are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Mac and Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.