LogMeIn Security:

An In-Depth Look

Table of Contents

Remote Access Axioms 3
LogMeIn Architecture 5
LogMeIn Security Mechanisms 7
Intrusion Detection 13
Data Forwarding 18
UDP NAT Traversal 19
Software Updates and Gateway Security 21
Conclusion 22

Author Márton Anka, Chief Technical Officer, LogMeIn, Inc.

Abstract

This paper provides an in-depth look at the security features of LogMeIn.com's remote access product, LogMeIn. We at LogMeIn.com do not believe in security through obscurity, nor do we expect our customers to blindly accept our claims to important security features, such as end-to-end encryption. By publishing the details on how the security mechanisms work and interoperate in our products, we are also inviting the public to scrutinize our efforts.

Audience

This document is technical in nature and is aimed at network engineers or network designers. Reading this paper can help the reader perform the necessary threat analysis before deploying our product.

Terminology

In the LogMeIn architecture, there are three entities that take part in every remote access session. The "client" or the "user" is the person or software accessing a remote resource. The "host" or the "server" is the computer being accessed, or the LogMeIn host software on this computer. The "gateway" is the LogMeIn service that mediates traffic between the client and the host.

Design Fundamentals

LogMeIn was designed to allow secure remote access to critical resources over an untrusted network. During the development of the software, security considerations always prevailed over usability concerns.

Remote Access Axioms

Everything Is a Target

As the penetration of broadband Internet connections increases, more and more computers are online 24/7. Most of these computers are operated by home users, and have gaping security holes, such as unpatched vulnerabilities and a lack of proper passwords.

The greatest weakness is, however, the user himself. The extremely quick penetration of so-called email viruses illustrates the lack of security-consciousness and the gullible nature of most Internet users. Email viruses, of course, are email attachments that are better classified as Trojan horses. They spread so quickly because users are surprisingly willing to violate fundamental rules when handling untrusted content. If the users themselves are responsible for infecting their computers with Trojans, how can you trust them to properly secure their systems against direct attacks?

Even competent network administrators can slip up and forget to install a patch or two, which, in the worst-case scenario, can allow attackers to run arbitrary code on the affected systems. Nothing demonstrates this better than the rapid spread of the Microsoft SQL Server worms in 2003. Both MSBlaster and Slammer infected a great number of computers, and generated such excessive amounts of network traffic - with the help of the exponentially growing number of infected hosts - that users could perceive a slowdown in Internet access even on uncompromised networks.

Worms like MSBlaster or Slammer were poorly written. Their spread rate was far from optimal, and they did not cause data loss or theft on a significant scale. Their creators exploited a widely known vulnerability in Microsoft SQL Server – a fix was available for several weeks before the first worm attack struck. In other words, they qualify as a very tasteless joke; one that undoubtedly caused many problems, but whose effect was nowhere near as disastrous as it could have been. One can only imagine what skilled hackers with malicious intent are capable of when they have a lucrative target.

Remote Access and Security

It is easy to see that many computers connected to the Internet are extremely vulnerable, even without installing a remote access product. Remote access products are perceived as high risk factors, but mainly for psychological reasons. When a user first sees a remote access solution in action, their first negative reaction is usually with regard to the security implications. This is perfectly normal, and, in fact, desirable. The real problem is that users do not immediately see the threat inherent in other network-enabled applications, such as an email client, a web server or the operating system itself.

All modern operating systems include some sort of remote access solution by default. Windows, for example, ships with Microsoft's Remote Desktop as a simple remote administration interface. Even OpenBSD, the Unix variant which is usually regarded as the most secure operating system available, includes SSH, which, again, is a simple and secure application that allows command-line access over a network connection to the remote computer.

In essence, a well-chosen and well-configured remote access solution does not carry any additional risks. If a network manager can keep a network secure using a reliable remote access software package, such as LogMeIn, productivity can be increased and costs reduced without any adverse effects on network security.

LogMeIn Architecture

Before explaining the exact security mechanisms employed by LogMeIn, it is necessary to give a quick introduction to the solution architecture.

There are three key components to any remote access session. The roles of the client and the host should be straightforward – the third component is the LogMeIn gateway.

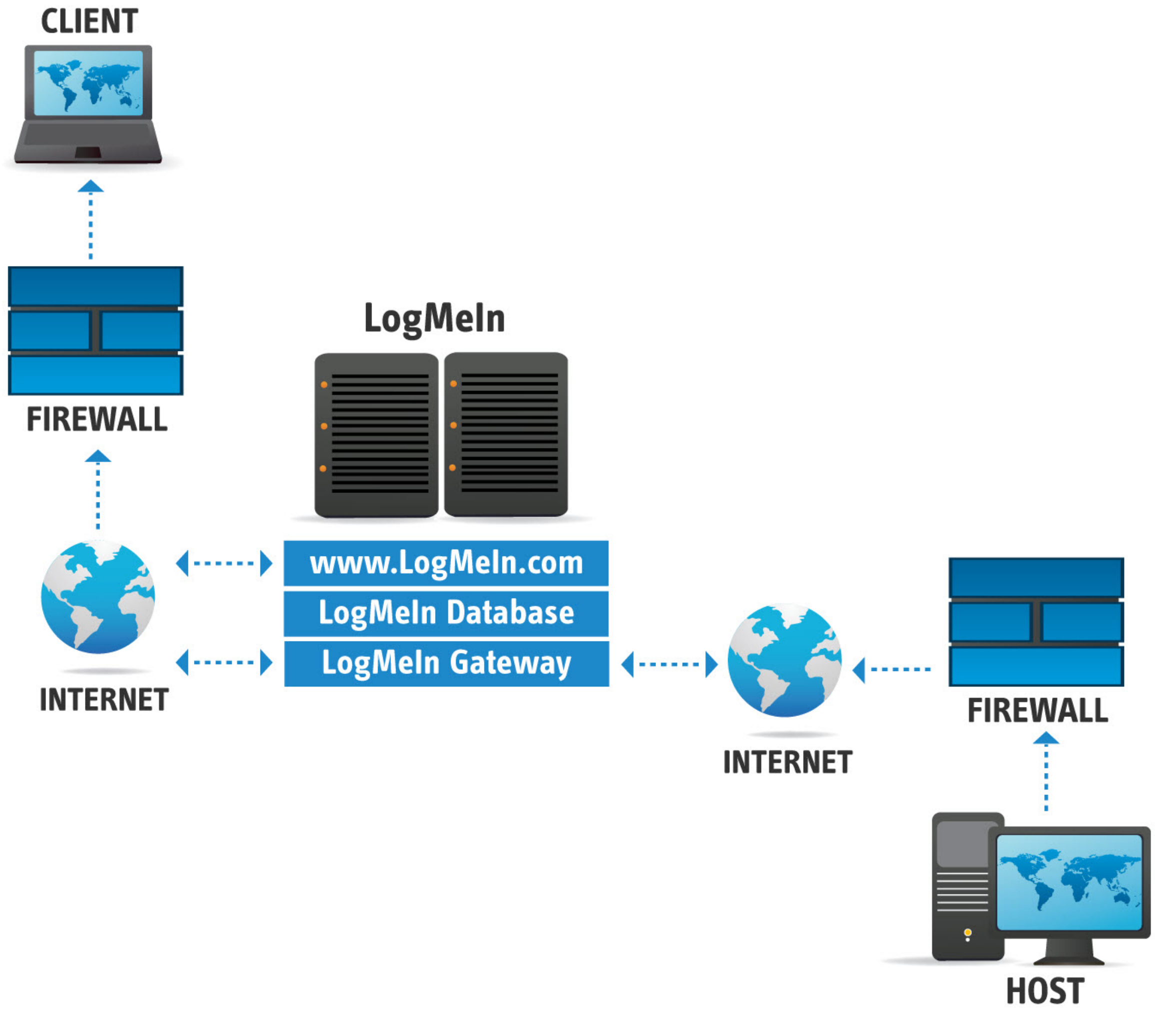

Figure 1. LogMeIn Architecture

The LogMeIn host in Figure 1 maintains a constant SSL-secured connection with one of the LogMeIn gateway servers in one of our physically secure datacenters. This link is initiated by the host and firewalls treat it as an outgoing connection, like secure web-browsing traffic.

The client browser establishes a connection to LogMeIn and authenticates itself. Based on the client's identity, it is authorized to exchange data with one or more hosts (the hosts belonging to the user's account).The gateway then forwards the subsequent encrypted traffic between the client and the host. It is worth noting that the client will also need to authenticate itself to the host – the gateway mediates the traffic between the two entities, but it does not require that the host implicitly trust the client. Once the host has verified the client's identity and authorized the client to access the computer the actual remote access session begins.

The benefit of using the gateway, instead of establishing a direct link between the client and the host, is that either the client or host (or both) can be firewalled. The LogMeIn gateway ensures that users do not need to configure firewalls.

LogMeIn Security Mechanisms

When users think of Internet data security, they are usually concerned about data encryption – to the point where security is measured in the length of the encryption key used. However, encryption and decryption, while being very important, are fairly trivial tasks compared to the other challenges faced by designers of secure systems. As you will see, data encryption is just one of the main goals set forth by the designers of LogMeIn.

Authentication of the Gateway to the Client

First and foremost, when a user connects to a LogMeIn installation via a gateway – the "server" – they need to be 100% positive that the computer they are about to exchange data with is really the one to which they intended to connect.

Suppose that an attacker poses as the server towards the user, and it poses as the user towards the server. The attacker, in this case, can sit between the two parties while reading, or possibly modifying, the data in transit. This is known as a "Man in the Middle", or MITM attack and is especially hard to protect against.

LogMeIn utilizes SSL/TLS certificates to verify server identities and thus protect against MITM attacks. When a connection is made, the server's certificate is verified. If the certificate was not issued by a certifying authority the user has chosen to trust, a warning will be presented. If the certificate was issued by a trusted certifying authority, but the hostname in the URL does not match the hostname included in the certificate, a different warning will be presented.

If the server passes these verifications, then the user's browser generates a "Pre-Master Secret" or PMS, encrypts it with the server's public key contained within its certificate, and sends it to the server. As ensured by the use of public key cryptography, only the server that holds the corresponding private key can decrypt the PMS. The PMS is then used to derive the Master Secret by both the user and the server, which, in turn, will be used to derive initialization vectors and session keys for the duration of the secure session.

In short, the above ensures that the user is establishing the connection with the server, and not with a third entity. Should a MITM attack be attempted, either one of the security warnings will be triggered or the PMS will be unknown to the MITM, effectively rendering the attack impossible.

(Suggested reading: *SSL and TLS: Designing and Building Secure Systems* by Eric A. Rescorla)

Authentication of Users to the Gateway

LogMeIn users must be authenticated by both the gateway and the host. An email address and password verification is performed whenever a user logs on to the LogMeIn website. Users are also advised to enable one or more of LogMeIn's extra security features to strengthen this authentication step.

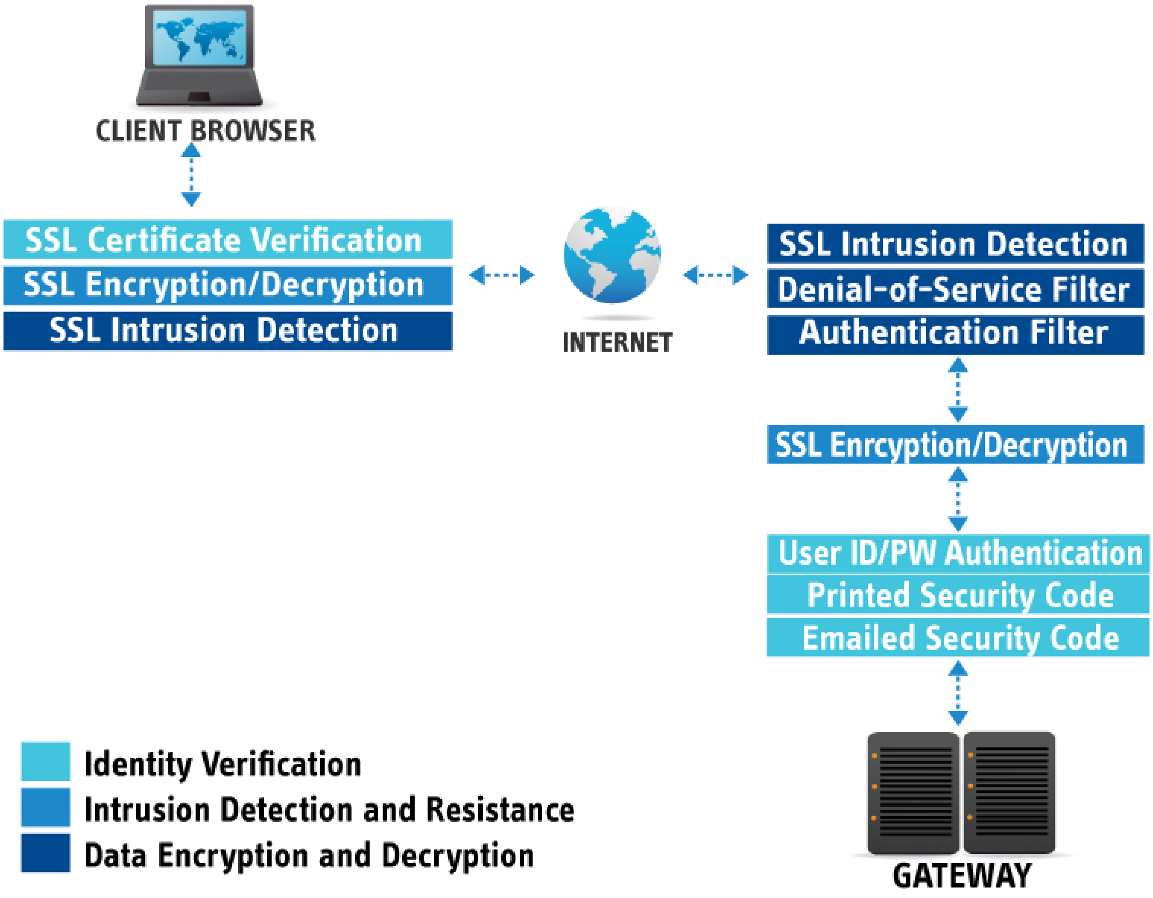

Figure 2. Authentication between Users and the Gateway

Printed Security Codes

One extra security feature is a sheet of printed security codes. When the user enables this feature, he is required to print out a list of nine-character random passwords generated by the gateway. Each time a LogMeIn user logs in to his account at www.LogMeIn.com, he will be prompted to enter one of the security codes from the list in order to gain access to his account. Each code can be used only once. Before the user runs out of printed security codes, he is required to print another sheet. This invalidates any previously unused security codes.

How to enable printed security codes.

1. Login to your LogMeIn account.
2. Click your email address and then **Account**.
3. On the **Security** tab, select the **Printed Security Code** option.
4. Generate and print a list of security codes.
5. Click **Save**.

Emailed Security Codes

Another way to secure your LogMeIn account is to use the Emailed Security Code feature. Each time you log in to your account at LogMeIn.com you will be sent an email containing a security code that you must then enter in the appropriate dialog before you can access your account. Each code can be used only once.

To make use of this technology, the receiver of the security code should be using a wireless device. When this feature is turned on and the user authenticates successfully with his email address and password to the LogMeIn gateway, a pass code is generated and sent to the wireless email address. The user receives this pass code in an email and enters the code into the form provided by the gateway - thus proving that he is in possession of the device. This password expires a few minutes after it has been generated or after it is used, whichever occurs first.

How to enable emailed security codes

1. Login to your LogMeIn account.
2. Click your email address and then **Account**.
3. On the **Security** tab, select the **Emailed Security Code** option.
4. Enter your email address in the field provided.
5. Click **Save**.

Account Audit

Use the Account Audit feature to keep track of activity in your LogMeIn account. Select events for which you want to receive automatic email notification, such as login attempt failure or password changes. Notifications will be sent to the specified email addresses (for multiple recipients, separate email addresses with a semi-colon). Note that some account events are turned on by default and cannot be disabled.

How to enable the account audit feature

1. Login to your LogMeIn account.
2. Click your email address and then **Account**.
3. On the **Security** tab under **Account Audit**, enter your email address in the field provided.
4. Select events for which you want to receive automatic email notification.
5. Click **Save**.

Authentication of the Gateway to the Host

The gateway must prove its identity to the host before it is trusted with access codes. The host, when making a connection to the gateway, checks its SSL certificate to make sure it is connecting to one of the LogMeIn gateway servers. This process is very similar to how a gateway authenticates a client.

Authentication of the Host to the Gateway

The gateway verifies the host's identity when it accepts an incoming connection using a long unique identifier string. This string is a shared secret between the two entities and is issued by the gateway when the host is installed. This unique identifier is only communicated over an SSL-secured channel, and only after the host has verified the gateway's identity. Figure 3 illustrates how the host and the gateway authenticate each other before a host is made accessible to the client.

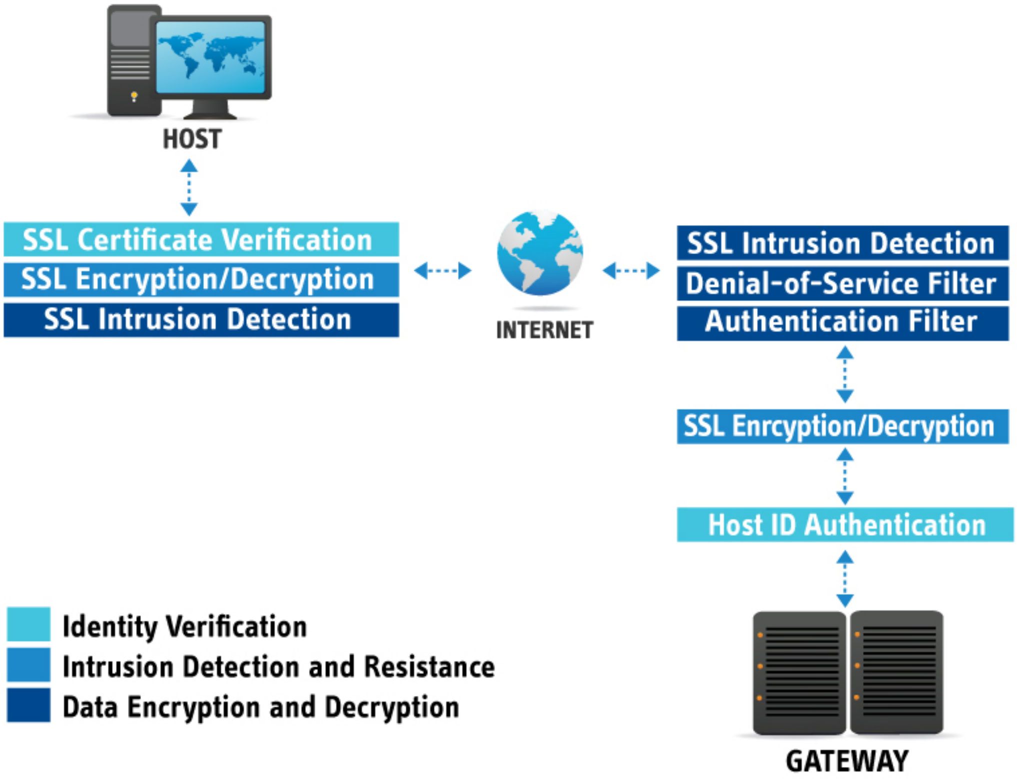

Figure 3. Host and Gateway Authentication

Data Encryption

The SSL/TLS standard defines a wide choice of cipher suites such as RC4 and 3DES, and some implementations offer more advanced suites that include AES as well. RC4 operates on 128 bit keys, 3DES uses 168 bit keys. AES can utilize 128 or 256 bit keys. The client and the host agree on the strongest cipher possible. This is done by the client sending the host a list of ciphers it is willing to use, and the host choosing the one it prefers from this list.

The SSL/TLS standard does not define how the host should choose the final cipher. In LogMeIn, the host simply selects the strongest available cipher suite that the client has offered.

This method allows both the client and the host to decline the use of specific data-encryption algorithms without the need of updating both components, should an algorithm be deemed as broken or insecure by research.

Intrusion Detection

LogMeIn provides two layers to detect intrusion attempts: SSL/TLS and LogMeIn Intrusion Filters.

SSL/TLS

The first layer of intrusion detection is provided by SSL/TLS to ensure that the data have not changed in transit. This is achieved by the following techniques:

Record Sequence Numbering

Record Sequence Numbering means that SSL/TLS records are numbered by the sender and the order is checked by the receiver. This ensures that an attacker cannot remove or insert arbitrary records into the data stream.

Message Authentication Codes

Message Authentication Codes (MACs) are appended to every SSL/TLS record. This is derived from the session key (known only to the two communicating parties) and the data contained within the record. If MAC verification fails it is assumed that the data were modified in transit.

Cipher Block Chaining

The cipher suites preferred by LogMeIn also utilize Cipher Block Chaining (CBC mode); meaning that every SSL/TLS record will depend on the contents of the previous record. In this mode, the input to the cipher is not only the current plaintext record, but the previous one as well. This again ensures that packets cannot be inserted or removed from the data stream.

For more information about SSL/TLS intrusion detection, see *SSL and TLS: Designing and Building Secure Systems* by Eric A. Rescorla.

LogMeIn Intrusion Filters

The second layer is provided by LogMeIn itself, and is comprised of three intrusion filters

IP Address Filter

When LogMeIn receives a connection request from a client, it first checks its list of trusted and untrusted IP addresses and possibly denies the connection. An administrator can set up a list of IP addresses within LogMeIn that are either allowed or denied to establish a connection to the selected host (for example, designate the internal network and another administrator's home IP address as allowed).

Denial of Service Filter

A Denial of Service Filter rejects connections if the IP address the request is coming from has made an excessive number of requests without authentication within the observation time window. This is done to protect against someone overloading the host computer by, for example, automatically and very quickly requesting the login page over and over again.

Authentication Filter

If the user made an excessive number of failed login attempts, the Authentication Filter rejects the connection. The Authentication Filter is in place to prevent a potential intruder from guessing an account name and password.

How to set filters on a LogMeIn host

1 Access the host preferences from either the host or the client:

- From the host, open LogMeIn and follow this path:
  Options > **Preferences** > **Security**
- From the client, connect to the host **Main Menu** and follow this path:
  Preferences > **Security**

2 Under **Intrusion Control**, click **Edit Profiles** to begin creating a filter profile.

For details, see the LogMeIn Pro2 User Guide, Free User Guide, or Central User Guide.

Authentication and Authorization of Users to the Host

After being granted access by the previous layers, the user must prove his identity to the host. This is achieved by a mandatory OS-level authentication step.

The user must authenticate himself to the host using his standard Windows or Mac username and password. The host will usually pass this request on to the relevant domain controller. This step not only validates the user's identity, but also ensures that network administrators can control who is able to log in to a specific host.

Personal Password

Personal Password is another optional security measure that can be set up on the LogMeIn host. The user can assign a Personal Password to the host, which, like the OS-level password, is not stored or verified by the gateway. A difference between the operating system password and the Personal Password is that the host never asks for the complete Personal Password so the user never enters it in its entirety in any single authentication session. The user is usually prompted for three random digits of the Personal Password by the host after OS-level authentication has succeeded. If the user enters the correct digits (for example, the first, the fourth and the seventh) he is granted access.

How to set up a Personal Password

1 Access the host preferences from either the host or the client:

- From the host, open LogMeIn and follow this path:
  Options > **Preferences** > **Security**
- From the client, connect to the host **Main Menu** and follow this path:
  Preferences > **Security**

2 Under **Personal Password**, enter your personal password and then enter it again to confirm.

3 Click Apply.

LogMeIn and RSA SecurID

To add an extra layer of security over the simple username/password authentication, you can configure LogMeIn Pro^2 and LogMeIn Free to require RSA SecurID authentication. RemotelyAnywhere, the product that pioneered the technology in use by LogMeIn, was officially certified by RSA Security as SecureID Ready in 2003. Since that time, LogMeIn has continued to maintain the high level of security consistent with RSA technology.

For information on the RSA SecurID product, visit www.rsa.com.

For information on setting up this feature on a LogMeIn Pro^2 or LogMeIn Free host, visit the LogMeIn Knowledge Base.

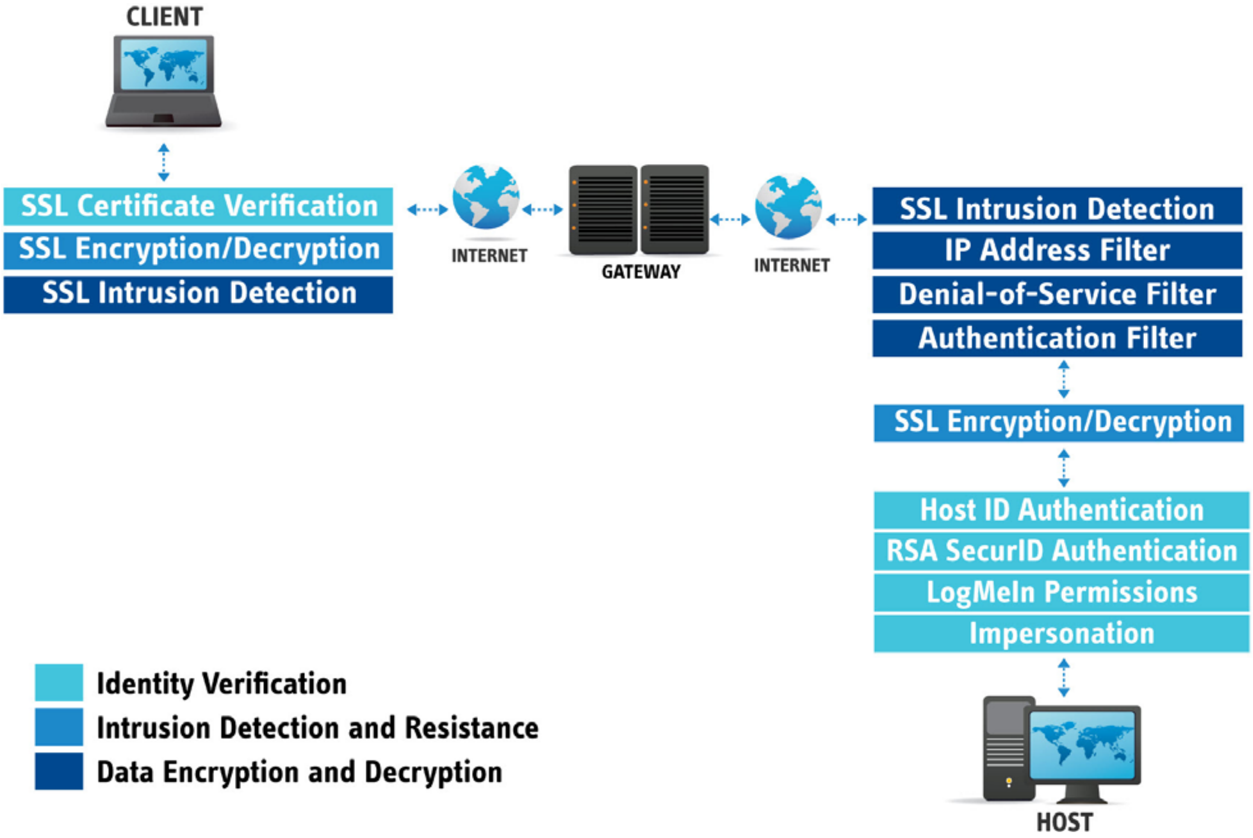

Figure 4: Authentication between Users and the Host

Authentication and Authorization of Users within the Host

Once LogMeIn has verified the user's identity using the above methods, it checks its own internal user database to see which internal modules the user is allowed to access.

System administrators can configure LogMeIn so that users with certain roles have access only to a subset of tools offered by LogMeIn; for example, the Helpdesk department can be configured to only view a computer's screen and performance data, but not actually take over the mouse and the keyboard or make any changes to the system configuration. Alternatively, the Sales department might be given full remote control access to their respective computers, but features such as performance monitoring and remote administration would be made unavailable to them.

Using the operating system access token obtained when the user was authenticated, LogMeIn impersonates the user towards the operating system while performing actions on their behalf. This ensures that LogMeIn adheres to the operating system's security model, and users have access to the same files and network resources as if they were sitting in front of their computer. Resources unavailable to users in Windows or OS X also remain unavailable via LogMeIn.

See "Controlling Who Can Access Your Host Computers" in the LogMeIn Pro^2 User Guide.

Auditing and Logging

LogMeIn provides extensive logging capabilities. A very detailed log of the events that occur within the software is kept in the installation directory. The most important events are also placed in the Windows application event log – these events include, for example, logon and logoff actions. The detailed log can also be sent to a central SYSLOG server.

See "How to View LogMeIn Event Log Files" in the LogMeIn Pro^2 User Guide for details.

For SYSLOG, see "Deployable Host Preferences for Logs and Session Recording" in the LogMeIn Central User Guide.

Data Forwarding

The gateway provides end-to-end encryption by forwarding encrypted data between the host and the client. If you are familiar with how SSL works, this might sound impossible; after all, the assumption is that since the client is confident that it is communicating with the gateway it is only the gateway that can decrypt the data sent by the client. This is a valid point, but LogMeIn made a few important changes to how SSL sessions are handled between the host and the gateway.

The first part of the SSL negotiation is performed between the gateway and the client. The gateway then passes the exchange on to the host, which re-negotiates the SSL session and agrees on a new session key with the client, thereby providing true end-to-end encryption.

When the traffic is relayed through the gateway, the client (browser) establishes an SSL session with the gateway using the gateway's certificate. The gateway transfers this SSL session's state (including the pre-master secret) to the host. After agreeing on a new session key, the host uses this session state to handle the rest of the SSL session directly with the client. As far as the client is concerned, the session is secured using the gateway's SSL certificate, but it is actually talking directly with the host, without the need for the gateway to decrypt and re-encrypt traffic.

A MITM attack is rendered impossible since both the host and the client verify the gateway’s certificate and the client uses its RSA public key to encrypt information that is used to derive the SSL/TLS Pre-Master Secret.

UDP NAT Traversal

It is important to explain how UDP NAT Traversal is used, especially since UDP is regarded as notoriously insecure. This is not entirely a misconception: if UDP is used as a communications medium, then security can be a serious problem, as UDP datagrams are easy to forge and the sender's IP address can be spoofed.

To counter this, LogMeIn.com does not use UDP as the communications medium itself with UDP NAT Traversal connections. UDP is relegated to the network layer, as defined by the ISO/OSI Network Model, with a TCP-like transport layer built on top of it, complete with flow control, dynamic bandwidth scaling and packet sequence numbering.

LogMeIn.com uses UDP instead of TCP packets (thereby effectively re-implementing a TCP-like transport layer) because most firewalls and NAT devices allow seamless two-way communication over a UDP transport as long as it is initiated from within the security perimeter, but they require significant reconfiguration for TCP and IP packets. After a reliable TCP-like stream is constructed from unreliable UDP packets, the stream is further protected by an SSL layer, providing full encryption, integrity protection and endpoint verification capabilities.

To set up a UDP NAT Traversal connection, both the client and the host send several encrypted UDP packets to the gateway. These packets are encrypted using a secret key shared by the gateway and the respective peer, and communicated over the pre-existing SSL connection. They are impossible to spoof.

The gateway uses these packets to determine the external (Internet) IP addresses of the two entities. It also tries to predict which firewall port will be used for communication when a new UDP packet is sent. It passes its findings down to the peers which then attempt to set up a direct connection. If the gateway can determine the port in use, the connection succeeds. The peers verify each other using another shared secret obtained from the gateway. An SSL session is established. The peers then communicate directly.

If a direct connection cannot be set up, the peers will connect back to the gateway over TCP and request that a forwarded, end-to-end encrypted session be used. This process takes only a few seconds and is transparent to the user. The only noticeable difference is the improved performance and low latency when a direct connection is in use.

For further details see US Patent no. 7,558,862.

Software Updates and Gateway Security

The LogMeIn host, based on user preferences, can semi-automatically or automatically update itself on the user's computer. The host software periodically checks the LogMeIn.com website for newer versions of the software. If a new version is found, it is automatically downloaded and a message is displayed to the user who can allow the update to take place. The download process uses at most 50% of the available bandwidth, therefore keeping interference with other networking applications to a minimum.

These software updates are digitally signed by LogMeIn.com with a private key that is not found on any of our Internet-connected systems. Therefore, even if the LogMeIn datacenters were compromised by attackers who then gain complete control over our servers, they would not be able to upload a rogue update and run arbitrary code on our users' computers. The most such a highly unlikely attack could accomplish is access to the LogMeIn logon screen on the customer's computer, which, even though it effectively bypasses the gateway security mechanisms, would still require that they enter valid operating system credentials to gain access to the computer. Brute-forcing the password is unfeasible, as the Authentication filter, by default, blocks the user's IP address after a few incorrect passwords.

In the event that our customers use the same password for the LogMeIn gateway and their computer, we do not store the actual LogMeIn passwords in our database; rather, we employ a one-way cryptographic hash and a per-account salt value to ensure that brute-forcing the password when in possession of the hash value is unfeasible given the computing resources available today.

Conclusion

A well-designed remote access solution can greatly increase productivity and provide a rapid return on investment. When deployment is done with care and LogMeIn's optional security features are utilized, the benefits greatly outweigh the risks.

References

SSL and TLS: Designing and Building Secure Systems by Eric A. Rescorla, Addison-Wesley Pub Co, 2001. ISBN: 0-201-61598-3

SSH, The Secure Shell: The Definitive Guide by Daniel J. Barrett, Ph. D., Richard E. Silverman, and Robert G. Byrnes, O'Reilly & Associates, 2001. ISBN: 0-596-00011-1

RSA Security's SecurID Product: http://www.rsa.com/node.aspx?id=1156

LogMeIn Support and Documentation: https://secure.logmein.com/support/