

Why is TeamWarden a safe place to keep your information?

The easy explanation

Each device has a public key and a private key that work together as a pair. Think of a locked box. The box can only be locked with the public key, and can only be unlocked with the private key. A box locked with the public key can't be unlocked with the public key.

Each device's public key is held on the server. Information that the device can view is locked into a box using the device's public key. This happens on the device that the information is entered on, before it's sent to the server. This information is then held on the server.

If someone breaks into the server and copies all the data, all they will have is a set of locked boxes. The keys to unlock the boxes are not on the server, but are on each device.

When the device wants to read the data, it copies the box to the device, and unlocks it with the private key. The device can then see the data. This only happens on the device – the data is never revealed on the server.

To identify itself to the server the device uses the same device key pair. Instead of sending a username and password, the device uses its public key to prove it is whom it claims, and the server can authorize access on this basis.

Because the key is stored on the device, you need to physically have the device to unlock the box with the data in it. Access can't be shared, or the password written down.

If you lose the admin key, we can't ever retrieve the information for you, because it's never on our server in a form that we can read.

The technical explanation

When an account is created an admin public/private key pair is created on the first device. The public key is uploaded to the server, and the key fingerprint (that uniquely identifies the public key) is used to identify the account. All new devices joining the account are given the fingerprint (typically via a QR code via email) and validate that the account fingerprint matches the public key of the account joined. If it does, this public key is then trusted as the admin key.

Each device creates a public/private key pair and the public key is registered on the server. A device starts in an untrusted state, where no information is released to it. The administrator views untrusted devices and can choose to sign

them with the private key (verifiable using the admin public key) as a trusted device. A challenge/response mechanism is provided that can be used to validate that the device in the database matches the device being validated.

Once a device is trusted, it can be assigned to have access to one or more teams. Each team's data is encrypted using the team symmetric key. A device is given access by encrypting the symmetric key with the device's public key. When it needs to read a team's data, it pulls the encrypted symmetric key from the server, decrypts it and uses it to read the data.

When writing data back to the server, we need to be sure we are writing it such that only trusted, assigned devices can read it. The first step is to obtain the key to encrypt the data. This is the same process as above – we read the encrypted team symmetric from the server, decrypt it using our private key, and then encrypt the data with that key. We need to defend against someone modifying the database and placing a key there that isn't the team key, but a different key. To do this the symmetric key is signed by the admin key. The device checks that the signature of the decrypted key matches the signature provided by the admin key – then it knows it can trust it to encrypt data with it.

The device authenticates to the server by sending an authorization header that signs the header with the device's private key. The authorization header includes a timestamp to defend against replay attack. The authorization header signature is validated against the device's public key to authenticate it.