SECURE REMOTE ACCESS OF COMPUTING DEVICE

Technical Field

Embodiments relate to a computer system and in particular to a computer system in which a first device is remotely accessed by a second device.

Background

A number of endpoint devices accessible over a network may be provided with computer executable instructions, such as software or firmware, for performing their required operations. For example, in the context of monitoring the usage of electricity or gas, the devices may take the form of smart meters that collect readings of electricity and gas usage and dispatch those readings in packets sent over the network to a headend system. To perform these operations, the smart meters comprise executable instructions held in the memory of the smart meter that provides control of the smart meter, enabling it to perform its operations of collecting measurement data and reporting the measurement data to the headend system.

At certain times, there may arise a need to permit access to one or more endpoint devices or other intermediate device within the network, e.g. for troubleshooting or maintenance purposes. One option for obtaining such access is for an engineer to physically travel to and access the relevant endpoint device. Whilst requiring the engineer to be physically may provide a high level of security, it may also be time consuming and inefficient. An alternative approach is to enable remote access to a device by providing an open port on the device so that the device may function as a server for remote access by a client using, for example, the Secure Shell (SSH) protocol. However, providing a continuously open port on the device for allowing remote access may present a security risk.

Summary

There is a requirement for an approach for enabling remote access to a device that addresses the above mentioned security concerns.

According to first aspect, there is provided a computer system comprising a first device comprising at least one processor and a memory storing computer readable instructions, wherein the first device comprises an interface configured to receive a command to enable remote access to the device, the command comprising a signed token, wherein the at least one processor is configured to execute the computer readable instructions to: verify that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server; in response to the verification of the signed token, send a request to initiate a connection to a second device identified in the token; following initiation of the connection, receiving via the connection, a request for remote access to the first device; and accepting the request for remote access.

The first device (which could, for example, be an endpoint device or a device for interfacing with an endpoint device) is enabled to receive commands to initiate a connection to an identified second device only in response to verifying that the command is signed by a trusted authority. In this way, a malicious third party cannot cause the first device to connect to a selected device of its choice. Once the connection is established, that connection is then used by the second device to request remote access to the first device. In this way, the first device ensures that only trusted devices may obtain remote access.

In a second aspect, there is provided a computer implemented method comprising: receiving a command to enable remote access to a first device, the command comprising a signed token; verifying that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server; in response to the verification of the signed token, sending a request to initiate a connection to a second device identified in the token; following initiation of the connection, receiving via the connection, a request for remote access to the first device; and accepting the request for remote access.

The method may be implemented in the first device or in a system comprising the first device and further devices (e.g. the second device).

In some embodiments, the token comprises: an internet protocol address providing the identification of the second device; and a port number of the second device, wherein

the method comprises sending the request to the initiate the connection to the port number of the second device.

In some embodiments, the connection to the second device is a secure shell connection, wherein the remote access to the first device is via a reverse secure shell connection made using the secure shell connection.

In some embodiments, the method comprises following initiation of the connection, dispatching the request for remote access to the first device.

In some embodiments, the second device is a second server, wherein the method comprises: following the acceptance of the request for remote access, receiving and executing commands originating from a third device via the connection with the second server.

In some embodiments, the second server is a cloud based server.

In some embodiments, the method comprises prior to the reception at the second device of the request to initiate the connection: the third device transmitting a command to establish the second device as a server available for connecting to the first device.

In some embodiments, the method comprises: the third device, prior to the reception at the second device of the request to initiate the connection: transmitting a command to the second device to unblock a port of the second device to accept the connection.

In some embodiments, the method comprises at a third server: generating the token identifying the second device; providing the token to the first server for signing; subsequently, receiving the signed token from the first server; and providing the command to the first device.

In some embodiments, further comprising the first server signing the token using the private key of the first server.

In some embodiments, method comprises, at the first server: following reception of the token at the first server, verifying that at least one of an address and/or port of the

second device in the token matches at least one expected value for the address and/or port; and in response to verifying the match, signing the token.

In some embodiments, the method comprises the first device granting the remote access in accordance with an access level containing within the token.

In some embodiments, the method comprises causing the connection to be terminated in response to at least one of: expiry of a predefined time limit defined in the token; and reception of a terminate command at the first device.

In some embodiments, the method comprises upon determining that the connection is to be terminated in response to the terminate command, send to the second device, a log of commands executed on the first device via the remote access.

In some embodiments, the remote access of the first device comprises at least one: executing at the first device, commands received from a remote device; and accessing one or more files stored on the first device by the remote device, wherein the remote device comprises the second device or a third device in communication with the second device.

In some embodiments, accepting the request for remote access comprises forming as a tunnel within the connection, a further connection with the second device.

In a third aspect, there is provided a computer program comprising computer readable instructions, which when executed by at least one processor cause a method according to the second aspect or any embodiment thereof to be performed.

In a fourth aspect, there is provided a non-transitory computer readable storage medium storing the computer program of the third aspect.

Brief Description of Drawings

Features will become apparent to those of ordinary skill in the art by describing in detail example embodiments with reference to the attached drawings, in which:

Figure 1 shows an example network in which a headend system communicates with a plurality of endpoints;

Figure 2 illustrates an example computing system comprising processing circuitry and storage;

Figure 3 illustrates an example computing device comprising processing circuitry and storage;

Figure 4 illustrates an example system in which a device of that system is subject to remote access over a network;

Figure 5A illustrates the first part of a method for remotely accessing a device of the system;

Figure 5B illustrates a second part of the method for remotely accessing a device of the system;

Figure 6 illustrates an unsigned access token and a signed access token, the signed access being for initiating a connection to a remote server; and

Figure 7 illustrates an example system for remotely accessing multiple devices of that system.

Detailed Description

Embodiments are implemented in a system comprising a first device that is subject to remote access from a second device. The first device may an endpoint device in a network comprising a plurality of endpoints for collecting measurement data.

Reference is made to Figure 1, which illustrates a network 100 in which a control system 102 interfaces with a plurality of endpoints 112 to 122 via a number of further network devices 104, 106, 108. Together, the system 102 and network devices 104, 106, 108 form a data processing system 130 in which embodiments are implemented. In some embodiments, the system 102 is a headend system 102, the devices 104, 106, 108 are collectors 104, 106, 108, and the endpoints 112 to 122 are smart meters 112 to 122.

The endpoints 112 to 122 are shown divided into a plurality of sets 110a-c. Each of the sets 110a-c interfaces with a different one of the collectors 104 to 108 for the purposes of sending measurement data to the system 102. Each of the sets 110a-c may communicate with the system 102 in accordance with a different communication technology, such as radio frequency (RF), power-line communication (PLC), and cellular.

For example, each endpoint belonging to the first set 110a may send measurement data to the collector 104 in packets provided in accordance with a first communication protocol and receive commands from the collector 104 in packets provided in accordance with the first communication protocol. Similarly, each endpoint belong to the second set 110b may send measurement data to the collector 106 in packets provided in accordance with a second communication protocol and receive commands from the collector 106 in packets provided in accordance with the second communication protocol. Each endpoint belonging to the third set 110c may send measurement data to the collector 108 in packets provided in accordance with a third communication protocol and receive commands from the collector 108 in packets provided in accordance with the third communication protocol. When each collector 104, 106, 108 receives an item of measurement data from one of the endpoints 112 to 122 with which it is associated, the collector 104, 106, 108 dispatches that item of measurement data to the system 102.

Reference is made to Figure 2, which illustrates an example computer system 200. Any of the devices and system described herein (including system 102, network devices 104 to 108, or endpoints 112 to 122) may be provided according to the example computer system 200. The system 200 may comprise a server, back-end system, or the like.

The system 200 comprises at least one memory 210, 220, at least one data processing unit 230, 240 and an input/output interface 250. The at least one memory 210, 220 comprises a random access memory 210 and at least one hard drive 220. The memories 210, 220 store computer executable code which, when executed by at least one data processing unit 230, 240, perform the steps described as being performed by the system 200. The memories 210, 220 may be used for storing data associated the control application. In particular, the memories 210, 220 may be used for storing a log file comprising data logged by the control application.

The at least one processor 230, 240 communicates with the memories 210, 220 to load instructions for execution, load data for processing, and store the results of processing that data. At the interface 250, the system 200 sends and receives messages. The messages received at the interface 250 comprise data for processing by the at least one processor 230, 240. The messages sent from the interface 250 are created by the at least one processor 230, 240.

As an alternative to the server type system 200, some of the devices and systems described herein may be provided as a different type of device having a user interface.

Reference is made to Figure 3, which illustrates an example of such a computing device 300. The device 300 may be a mobile user equipment (UE), a personal computer (PC), a terminal or workstation, or some other form of device.

The device 300 comprises an interface 340 over which it sends and receive signals. The interface 340 may be a wired or wireless interface. For instance, the interface 340 may comprise a wired interface for connection to a wired network (e.g. a local area network and/or the internet). Alternatively or in addition, the interface 340 may comprise transceiver apparatus configured to send and receive communications over a radio interface. The transceiver apparatus may be provided, for example, by means of a radio part and associated antenna arrangement. The antenna arrangement may be arranged internally or externally to the system 300.

The system 300 is provided with at least one data processing entity 315, at least one random access memory 320, at least one read only memory 325, and other possible components 330 for use in software and hardware aided execution of tasks it is designed to perform, including control of, access to, and communications with access systems and other communication devices. The at least one random access memory 320 and the hard drive 325 are in communication with the data processing entity 315, which may be a data processor. The data processing, storage and other relevant control apparatus can be provided on an appropriate circuit board and/or in chipsets. A user controls the operation of the system 300 by means of a suitable user interface such as key pad 310, or by voice commands. A display 305 is included on the system 300 for displaying visual content to a user. The system 300 may also comprise a speaker for providing audio content.

The storage of the system 300 (i.e. the random access memory 320 and the hard drive 325) is configured to store computer readable instructions for execution by the data processor 315 to perform the data processing functions described herein as being performed by a system or device.

Reference is made to Figure 4, which illustrates a computer system 400 comprising a first device 410. The first device 410 may be a device that, during its operation, encounters a maintenance or troubleshooting issue that requires remote access. The first device 410 may be an endpoint device, such as a smartmeter. The first device 410 may be any of the endpoints 112 to 122 discussed above with respect to Figure 1. The first device 410 may alternatively be an intermediary device in a system, e.g. the first device 410 may be one of the collectors 104 to 108 discussed above with respect to Figure 1. The first device 410 is also referred to herein as the remotely accessed device 410, since it is the subject of remote access by a second device 430. The first device 410 may an internet of things (IoT) device 410, which is connected to a Wi-Fi access point via which it connects to the second device 430 over the internet.

The system 400 comprises the second device 430 to which the remotely accessed device 410 is configured to connect in response to an appropriate command. In the example of Figure 4, the second device 430 is shown as a server 430, which may be created on demand in the cloud. The server 430 may be referred to as an engineering access tool (EAT) server 430. The remotely accessed device 410 initiates a connection (which may be referred to herein as the 'initial connection' or 'original connection') with the second device 430. The connection may be a secure shell (SSH) protocol connection, which may use TCP/IP. Once the initial connection is established, the second device 430 may initiate a further connection with the remotely accessed device 410, which is a tunnel within the original connection. The further connection permits remote access by the second device 430 to the remotely accessed device 410. The further connection may also be a secure shell (SSH) protocol connection, which is tunnelled within the original SSH connection. The remote access of the second device 430 to the remotely accessed device 410 may, therefore, achieved by reverse SSH.

In establishing the initial connection to the second device 430, there is a client-server relationship between the first device 410 and the second device 430, i.e. the first device 410 acts as a client whilst the second device 430 acts as a corresponding server. By establishing the initial connection in this manner, the remotely accessed device 410 does not need to have an open port (which may represent a security risk) for listening to the connection requests. The second device 430 may have associated firewalls that reduce the security risk associated with having an open port on that device 430.

As noted, the second device 430 may be a server, in which case the server 430 may interface with a third device 440 belonging to the system 400 and enable the third device 440 to have remote access to the remotely accessed device 410. The third device 440 may, for example, be a personal computer of an engineer who is required to remotely access the first device 410 to perform maintenance or trouble shooting. The third device 440 may be referred to herein as the engineer device 440. When establishing the remote access to the remotely accessed device 410, the engineer device 440 may login to the server 430 and cause the server 430 to create the further connection (which is a tunnel within the initial connection) to the device 410 for allowing remote access.

The system 400 further comprises a control system 420, which provides the connect command to the remotely accessed device 410 to cause it to establish the initial connection with the second device 430. The control system 420 may be the head end system 102 shown in Figure 1. The control system 420 may be provided by one or more servers and may be referred to as the 'third server'.

The system 400 further comprises a server 450, which may be referred to as a 'first server' 450 or as a signing server 450. The signing server 450 is configured to interface with the control system 420 and to receive an access token from the control system 420 for signing and to provide the signed token to the control system 420. The signing server 450 stores a private key for providing the signed token. The signing server 450 and the control system 420 may be controlled by different entities, such that the entity responsible for the control system 420 does not have access to the private key stored in the signing server 450. In the example in which the control system 420 is a head end system, the control system 420 may be controlled by a utility company, whilst the signing server 450 is controlled by a manufacturer of the device 410. By maintaining control of the private key, the manufacturer may prevent malicious third parties from gaining accessed to its devices, such as device 410. The control system 420 and the signing server 450 may communicate via a pre-defined channel via which the unsigned access token is sent to the signing server 450 and the signed access token is returned to the control system 420.

Each of the devices/systems 410, 420, 430, 440, 450 shown in Figure 4 may be provided in accordance with either of the example devices 200, 300 discussed above, and reference herein to operations performed by any of the devices/systems 410, 420, 430,

440, 450 are understood to refer to operations performed in response to at least one processor (e.g. processors 230, 240, 315, 330) of the respective device/system executing computer readable instructions stored in memory of the device/system to cause those operations to be performed.

Reference is made to Figure 6, which illustrates examples of an unsigned token 600 and an example of a signed token 610. The unsigned token 600 is generated by the control system 420. The unsigned token 600 comprises an identifier 602 of the first device 410, which may take the form of a serial number 602. The identifier 602 of the first device 410 prevents the token from being used to enable remote access to any other device than the one identified in the token 600. The remotely access device 410 upon receipt of the token 610 may determine to enable remote access only if the identifier 602 identifies that device 410.

The token 600 further comprises an address 604 of the second device 430, which is used by the remotely accessed device 410 to initiate the initial connection. The address 604 may be an IP address. The token 600 further comprises a port number 606, which identifies a port of the second device 430 to which the remotely accessed device 410 is configured to connect. The token 600 further comprises a device access level 608, which indicates the level of access provided to the device 410 when it is subject to remote access by the device 430. The level of access may determine the software and data that can be accessed on the device 410 and the commands that may be executed. The token 600 further comprises an access expiration time 610, which indicates the amount of time that the remote access to the device 410 is permitted before the initial connection (and hence the remote access to the device 410) is terminated by the device 410.

When the token 600 is subject to signing by the signing server 450, the signed token 610 is produced. The signed token 610 may comprise the same fields 602 to 610 as in the unsigned token 600, but further comprises the signature 612, which is produced by the signing server 450 applying the private key to the token 600. The signing server 450 may produce the signature 612 by applying the private key to all of the fields 602 to 610 of the token 600.

Reference is made to Figure 5A, which illustrates the first part 500 of a method according to embodiments. It would be appreciated that not all of the steps of the method are essential, but that in some embodiments one or more steps may be omitted.

In S502, an indication of remote access being required is received at the engineer device 440. This indication that remote access is required may be provided by the control system 420 or by another system controlled by the same entity as the control system 420. The indication may be provided from a customer relationship management (CRM) system (not shown in Figure 4) at which an issue requiring remote access has been logged. The indication received at S502 may be a log of one or more issues associated with the device 410 that require attention, e.g. troubleshooting or maintenance.

In S504, in response to the indication received at S502, the engineer device 440 may transmit a request to server 430 to cause the server 430 to be enabled in the cloud for receiving connection requests from the device 410. When the server 430 is initially setup at S504, the relevant port for receiving connection requests from the device 410 may be blocked until a further command to unblock the port is received from device 440.

At S506, the engineer device 440 sends a request to the control system 420 to cause the control system 420 to generate the access token. As part of the request, the engineer device 440 may dispatch to the control system 420, indications of fields belonging to the access token that identify the second device 430. In particular, the engineer device 440 may provide the IP address 604 and port number 606 to the control system 420. The engineer device 440 may also supply one or more of the other fields 602, 608, 610 of the token.

At S508, the control system 420 may create the access token. The control system 420 may create the access token based on one or more fields (e.g. fields 604, 606) received from the engineer device 440. The control system 420 may also store information, or retrieve information from an associated storage, that provides one or more of the fields. For example, the control system 420 may retrieve the serial number of the device 410, so as to provide the field 602.

At S510, the control system 420 may provide the generated access token to the signing server 450.

At S511, before signing the received token, the signing server 450 may verify the token. This verification may comprise checking the address and/or port (e.g. the fields 604, 606) to verify that the address and/or port match the values of the second device 430. In embodiments, the signing server 450 is under the control of the same entity (i.e. the manufacturer of the device 410), and so has access to the address and port values for the server 430 that was created for carrying out the remote access to the device 410. If the received token is verified by the server 450, the method proceeds to S512.

At S512, the signing server 450 applies its private key to the token received from the control system 420 to add a signature to the token. The signature may be calculated over all or some of the fields of the token.

At S514, the signing server 430 returns the signed token to the control system 420.

At S516, in response to the receipt of the signed token, the control system 420 may transmit an indication that it has received the signed token and is ready to transmit the connect command to the device 410. In response, the engineer device 440 may unblock the port of the second device 430, so that the second device 430 is then operable to receive connection requests from the device 410. An indication that the port is unblocked may be returned to the control system 420, which in response proceeds to S518.

At S518, the control system 420 transmits a command (the 'connect command') to cause the first device 410 to initiate the initial connection to the second device 430. The connect command is received at an interface (e.g. interface 250, 340) of the first device 410.

At S520, the first device 410 verifies the signature that is included within the access token. The first device 410 performs the verification using a public key that corresponds to the private key of the signing server. The first device 410 may be configured to store the public key before the connect command is received, so that it is available when required.

Reference is made to Figure 5B, which shows the second part 550 of the method. After step S520, the method proceeds to S522 at which the first device 410 determines whether or not the decrypted signature matches the one or more fields of the token from

which the signature was produced using the private key. If the fields do not match, this is an indication that the token in the command may have been signed by an untrusted entity. Therefore in this case, at S524, the first device 410 determines not to initiate the initial connection. If, on the other hand, the fields do match, the method proceeds to S526 at which the first device 410 initiates the initial connection. The initial connection may be an SSH connection, in which case the first device 410 enables an SSH tunnel to be formed between the two devices 410, 430 so that, at S528, a further SSH connection is tunnelled within the initial SSH connection, so as to achieve remote access to the device 410.

The second device 430 receives the connection request (sent at S526) from the remotely accessed device 410. The connection request may be an SSH request. The second device 430 authenticates the connection request at S527 prior to accepting the connection request. The authentication may be carried out using a password, where the password stored by the second device 430 is provided by the device 410 when sending the request to initiate the initial connection. Alternatively, the authentication may be performed using a private-public key pair. In this case, the first device 410 may store a private key and the second device 430 may store a corresponding public key. Configuring the second device 430 in advance of the connection request being sent S526 may include installing an operational PKI (public key infrastructure) chain on the device 430 for SSH authentication and installing an SSH client certificate on the device 430 using a PKI chain, which is known to the remotely accessed device 410. When the first device sends the request to initiate the initial connection, this request may comprise a message signed with the private key of the device 410, which the second device 430 may verify using the public key.

Once the connection request is authenticated, at S527, the second device 430 accepts the connection request, so that the initial connection is established. The server 430 stores the port that the device 410 is using for initial connection to enable it to establish a further connection to the device 410. Following establishment of the connection, at S528, the second device 430 issues a request to establish the further connection within the initial connection (i.e. using the initial connection to provide a tunnel) established at S526. The establishment of this further connection is achieved by the second device 430 sending a request for remote access (e.g. an SSH request) via the initial connection to the device 410, with that request being accepted by the device 410. Once the further

connection is established, the device 430 then has remote access to the first device 410 via the further connection.

When the second device 430 has access to the first device 410, the second device 430 may access data stored on the first device 410 and perform command line execution on the first device 410. The second device 430 may perform these operations at a level of access that may be indicated (e.g. by field 608) in the access token. When the device 410 receives requests to access data or to execute commands via the remote access connection from the second device 430, it may accept or reject such commands in dependence upon whether they are permitted by the access level in the access token that was received at S514.

As discussed, the second device 430 may be a server, which interfaces with a third device 440. In this case, the remote access to the device 410 may be initiated by a command from the third device 440 to the server 430, which causes the server 430 to establish the remote access. The third device 440 may then send commands to the server 430 for execution on the first device 410 by the established connection between the server 430 and first device 410 and may retrieve data from the first device 410 via the server 430.

The remote access by the second device 430 to the first device 410 may continue until the initial connection is terminated by the device 410. The device 410 may determine that the initial connection is to be terminated in response to receipt of a terminate command from the control system 420 or the expiry of a timer.

At S529, the remotely accessed device 410 determines whether or not a request to returns a log of the commands executed on the device 410 so far, as part of the remote access session, has been received at the remotely accessed device 410 from the control system 420. If so, the method proceeds to S530, at which the device 410 returns the log of the commands that have been executed on the first device 410 as part of the remote access session. This log is returned to the second device 430 from where it may be returned to the engineer device 440.

At S531, the remotely accessed device 410 determines whether or not a terminate command has been received at the remotely accessed device 410 from the control system 420. The control system 420 may determine to send a terminate command in response to an indication from the engineer device 440 that the remote access is to be terminated. If a terminate command is received from the control system 420, at S534 the device 410 terminates the initial connection to the second device 430, and so causes remote access from the second device 430 to cease. The remote access is not re-enabled unless a further connect command is received from the control system 420.

At S532, the remotely accessed device 410 determines whether or not a timer has expired. The time limit of the timer may be set based on the time limit field 610 in the access token, which specifies a certain amount of time for which remote access is permitted. Once this amount of time has expired, the method proceeds to S536 at which the device 410 terminates remote access.

The remote access to device 410 may continue until either the terminate command is received at the device 410 or the time limit for remote access is reached.

When the remote access is terminated at S534 in response to the terminate command, the first device 410 may return to the second device 430, a log of the commands that were executed on the first device 410 as part of the remote access session. This log may be returned to the engineer device 440.

In some embodiments, a system may be provided comprising multiple remotely accessed devices. In the case in which these devices are associated with the same entity - for example, the same company (e.g. a utility company) - the system may comprise the same control system 420 and the same second device 430. However, in the case that the remotely accessed devices 410a, 410b are associated with different entities, separate instances of the second device and control system may be provided.

Reference is made to Figure 7, which illustrates an example of a system 700 comprising remotely accessed devices 410a, 410b, which may be controlled by separate entities. Each of the devices 410a, 410b may interface and communicate with an associated control system 420a, 420b and an associated second device 430a, 430b in the manner described above with respect to Figure 4. To remotely access the devices 410a, 410b,

the engineer device 440 may establish respective servers 430a, 430b. Each of these servers 430a, 430b may connect to its associated device 410a, 410b by initiating a request for remote access through an initial connection with its associated device 410a, 410b. The control systems 420a, 420b may each provide access tokens to the signing server 450, which provides corresponding signed access tokens in return using its stored private key. The access token generated by the control system 420a comprises the address (e.g. the IP address) of the second device 430a, whilst the access token generated by the control system 420b comprises the address (e.g. the IP address) of the second device 430b.

In some of the above embodiments, it was been described that the second device 430 functions as a server 430, which enables a third device 440 (e.g. a personal computer of an engineer) connected to the server 430 to access the device 410. However, in some embodiments, the second device 430 could be an endpoint device that a user uses to remotely access the device 410.

Implementations of the subject matter and the operations described in this specification can be realized in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. For instance, hardware may include processors, microprocessors, electronic circuitry, electronic components, integrated circuits, etc. Implementations of the subject matter described in this specification can be realized using one or more computer programs, i.e., one or more modules of computer program instructions, encoded on computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer

storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

While certain arrangements have been described, the arrangements have been presented by way of example only, and are not intended to limit the scope of protection. The inventive concepts described herein may be implemented in a variety of other forms. In addition, various omissions, substitutions and changes to the specific implementations described herein may be made without departing from the scope of protection defined in the following claims.

Annex A

The present invention can also be defined by reference to the following clauses:

Clause 1. A computer system comprising a first device comprising at least one processor and a memory storing computer readable instructions,

wherein the first device comprises an interface configured to receive a command to enable remote access to the first device, the command comprising a signed token,

wherein the at least one processor is configured to execute the computer readable instructions to:

verify that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, send a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receive via the connection, a request for remote access to the first device; and

accept the request for remote access.

Clause 2. A computer system as claimed in clause 1, wherein the token comprises:

an internet protocol address providing the identification of the second device; and

a port number of the second device,

wherein the at least one processor of the first device is configured to execute the computer readable instructions to send the request to the initiate the connection to the port number of the second device.

Clause 3. A computer system as claimed in clause 1 or clause 2, wherein the connection to the second device is a secure shell connection, wherein the remote access to the first device is via a reverse secure shell connection made using the secure shell connection.

Clause 4. A computer system as claimed in any of clauses 1 to 3, further comprising the second device comprising at least one processor configured to execute

computer readable instructions to, following initiation of the connection, dispatch the request for remote access to the first device.

Clause 5. A computer system as claimed in any of clauses 1 to 4, wherein the second device is a second server,

wherein at least one processor is configured to execute the computer readable instructions to:

following the acceptance of the request for remote access, receive and execute commands originating from a third device via the connection with the second server.

Clause 6. A computer system as claimed in clause 5, wherein the second server is a cloud based server.

Clause 7. A computer system as claimed in clause 5 or clause 6, wherein the third device is configured to, prior to the reception at the second device of the request to initiate the connection:

transmit a command to establish the second device as a server available for connecting to the first device.

Clause 8. A computer system as claimed in any of clauses 5 to 7, wherein the third device is configured to, prior to the reception at the second device of the request to initiate the connection:

transmit a command to the second device to unblock a port of the second device to accept the connection.

Clause 9. A computer system as claimed in any of clauses 1 to 8, further comprising a third server comprising at least one processor configured to execute computer readable instructions to:

generate the token identifying the second device;

provide the token to the first server for signing;

subsequently, receive the signed token from the first server; and

provide the command to the first device.

Clause 10. A computer system as claimed in any of clauses 1 to 9, further comprising the first server comprising at least one processor configured to execute computer readable instructions to:

sign the token using the private key of the first server.

Clause 11. A computer system as claimed in clause 10, wherein the at least one processor of the first server is configured to execute the computer readable instructions to:

following reception of the token at the first server, verify that at least one of an address and/or port of the second device in the token matches at least one expected value for the address and/or port; and

in response to verifying the match, signing the token.

Clause 12. A computer system as claimed in any of clauses 1 to 11, further comprising the second device.

Clause 13. A computer system as claimed in any of clauses 1 to 12, wherein the at least one processor of the first device is configured to grant the remote access in accordance with an access level containing within the token.

Clause 14. A computer system as claimed in any of clauses 1 to 13, wherein the at least one processor of the first device is configured to:

cause the connection to be terminated in response to at least one of:

expiry of a predefined time limit defined in the token; and

reception of a terminate command at the first device.

Clause 15. A computer system as claimed in clause 14, wherein the at least one processor of the first device is configured to:

upon determining that the connection is to be terminated in response to the terminate command, send to the second device, a log of commands executed on the first device via the remote access.

Clause 16. A computer system as claimed in any of clauses 1 to 15, wherein the remote access of the first device comprises at least one:

execution by the at least one processor of the first device of commands received from a remote device; and

access to one or more files stored on the first device by the remote device,

wherein the remote device comprises the second device or a third device in communication with the second device.

Clause 17. A computer system as claimed in any of clauses 1 to 16, wherein accepting the request for remote access comprises forming as a tunnel within the connection, a further connection with the second device.

Clause 18. A computer implemented method comprising:

receiving a command to enable remote access to a first device, the command comprising a signed token;

verifying that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, sending a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receiving via the connection, a request for remote access to the first device; and

accepting the request for remote access.

Clause 19. A computer program comprising computer readable instructions, which when executed by at least one processor cause a method to be performed, the method comprising:

receiving a command to enable remote access to a first device, the command comprising a signed token;

verifying that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, sending a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receiving via the connection, a request for remote access to the first device; and

accepting the request for remote access.

Claims

1. A computer system comprising a first device comprising at least one processor and a memory storing computer readable instructions,

wherein the first device comprises an interface configured to receive a command to enable remote access to the first device, the command comprising a signed token,

wherein the at least one processor is configured to execute the computer readable instructions to:

verify that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, send a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receive via the connection, a request for remote access to the first device; and

accept the request for remote access.

2. A computer system as claimed in claim 1, wherein the token comprises:

an internet protocol address providing the identification of the second device; and

a port number of the second device,

wherein the at least one processor of the first device is configured to execute the computer readable instructions to send the request to the initiate the connection to the port number of the second device.

3. A computer system as claimed in claim 1, wherein the connection to the second device is a secure shell connection, wherein the remote access to the first device is via a reverse secure shell connection made using the secure shell connection.

4. A computer system as claimed in claim 1, further comprising the second device comprising at least one processor configured to execute computer readable instructions to, following initiation of the connection, dispatch the request for remote access to the first device.

5. A computer system as claimed in claim 1, wherein the second device is a second server,

wherein at least one processor is configured to execute the computer readable instructions to:

following the acceptance of the request for remote access, receive and execute commands originating from a third device via the connection with the second server.

6. A computer system as claimed in claim 5, wherein the second server is a cloud based server.

7. A computer system as claimed in claim 5, wherein the third device is configured to, prior to the reception at the second device of the request to initiate the connection:

transmit a command to establish the second device as a server available for connecting to the first device.

8. A computer system as claimed in claim 5, wherein the third device is configured to, prior to the reception at the second device of the request to initiate the connection:

transmit a command to the second device to unblock a port of the second device to accept the connection.

9. A computer system as claimed in claim 1, further comprising a third server comprising at least one processor configured to execute computer readable instructions to:

generate the token identifying the second device;

provide the token to the first server for signing;

subsequently, receive the signed token from the first server; and

provide the command to the first device.

10. A computer system as claimed in claim 1, further comprising the first server comprising at least one processor configured to execute computer readable instructions to:

sign the token using the private key of the first server.

11. A computer system as claimed in claim 10, wherein the at least one processor of the first server is configured to execute the computer readable instructions to:

following reception of the token at the first server, verify that at least one of an address and/or port of the second device in the token matches at least one expected value for the address and/or port; and

in response to verifying the match, signing the token.

12. A computer system as claimed in claim 1, further comprising the second device.

13. A computer system as claimed in claim 1, wherein the at least one processor of the first device is configured to grant the remote access in accordance with an access level containing within the token.

14. A computer system as claimed in claim 1, wherein the at least one processor of the first device is configured to:

cause the connection to be terminated in response to at least one of:

expiry of a predefined time limit defined in the token; and

reception of a terminate command at the first device.

15. A computer system as claimed in claim 14, wherein the at least one processor of the first device is configured to:

upon determining that the connection is to be terminated in response to the terminate command, send to the second device, a log of commands executed on the first device via the remote access.

16. A computer system as claimed in claim 1, wherein the remote access of the first device comprises at least one:

execution by the at least one processor of the first device of commands received from a remote device; and

access to one or more files stored on the first device by the remote device,

wherein the remote device comprises the second device or a third device in communication with the second device.

17. A computer system as claimed in claim 1, wherein accepting the request for remote access comprises forming as a tunnel within the connection, a further connection with the second device.

18 A computer implemented method comprising:

receiving a command to enable remote access to a first device, the command comprising a signed token;

verifying that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, sending a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receiving via the connection, a request for remote access to the first device; and

accepting the request for remote access.

19. A non-transitory computer readable medium storing a computer program comprising computer readable instructions, which when executed by at least one processor cause a method to be performed, the method comprising:

receiving a command to enable remote access to a first device, the command comprising a signed token;

verifying that the signed token has originated from a first server by verifying the signed token using a public key corresponding to a private key of the first server;

in response to the verification of the signed token, sending a request to initiate a connection to a second device identified in the token;

following initiation of the connection, receiving via the connection, a request for remote access to the first device; and

accepting the request for remote access.