



Intel[®] Technology Journal

Interoperable Home Infrastructure

Home Network Security

Home Network Security

Carl M. Ellison, Corporate Technology Group, Intel Corporation

Index words: firewall, UPnP, 802.11, wireless, VPN, security, home networking

ABSTRACT

Home computers that are connected to the Internet are under attack and need to be secured. That process is relatively well understood, even though we do not have perfect solutions today and probably never will. Meanwhile, however, the home computing environment is evolving into a home network of multiple devices, which will also need to be secured. We have little experience with these new home networks and much research needs to be done in this area. This paper gives a view of the requirements and some of the techniques available for securing home networks.

INTRODUCTION

First, there was a single Personal Computer (PC) in a few homes with no connection to the outside world. Now, we have computers in most homes and most have Internet connections to the outside world. The next step, already happening, is not one computer but rather a large network of devices in a home. Some of these are mobile devices, which will be brought into the home by guests, friends, hired employees, maintenance personnel employed by service providers, and other strangers.

As these changes happen, the security needs of the home user also change. In the days of the disconnected single PC, the primary security threat was from virus contamination on floppy disks. With continuous connectivity to the Internet, many new attack channels have been opened (e-mail attachments, executable code or scripts fetched from Web pages, active penetrations at lower networking levels, etc.), while floppies have all but disappeared, closing that older channel. To the extent that these existing threats are understood, there are products available to help home users defend themselves against them.

However, the future home will have not one computer connected to the Internet but rather a network of many devices within the home, and that network might be connected to the Internet. In such an environment, the

potential for attacks is greatly increased. Since this is still in the future, there are no products to counter these attacks. This is therefore an area ripe for research and product development. This paper primarily addresses researchers and product developers considering this new environment.

We briefly address the present state of affairs regarding the security of home computers. Present security measures will continue to be valuable in the future and will continue to evolve. Security solutions are always evolving, as no solution remains adequate for long.

The bulk of this paper, however, discusses the new home environment, in which there are threats not only from outside but also from inside. Those threats are characterized, and security mechanisms that can be built into products to secure the home user against these threats are described.

In our conclusion we describe how security mechanisms built for the corporate environment have serious flaws when used in the home environment. We discuss Universal Plug and Play (UPnP^{*}), developed in response to the unique needs of the home environment.

SECURING THE EXISTING HOME NET

Any home computer connected to the Internet is in danger of being attacked. A broadband connection leads to probes preparatory to an attack every few minutes. A dial-up connection, behind the firewall of an Internet Service Provider (ISP), leads to attacks from machines that are behind the same firewall. In the author's experience with one ISP, probes came once or twice a week.

There exist many papers, both academic and practical, on how to use existing products to secure current home computers from attacks via the Internet. It is not the

* Other brands and names are the property of their respective owners.

purpose of this paper to reiterate that advice, but to summarize it:

1. Computer owners should have a firewall and allow no responses to any attempts to connect into the home from outside. A firewall must have external administration disabled, and any passwords with which it was shipped need to be changed to very secure, hard to guess, passwords. These passwords can be written down, because they are defending against network attackers rather than in-home attackers.
2. A computer should have a modern virus scanner, which is enabled to scan all inputs to the computer, as well as automatic updating of virus signature files, at least daily.
3. Computer owners should update operating systems and applications with the latest security patches and scan for new patches daily. These patches must be digitally signed, and therefore authenticated, as having come from the software vendor and not an attacker.
4. Security settings should be set to maximum on both browsers and e-mail agents.
 - a. E-mail agents should not allow incoming mail in HTML to be displayed if it accesses anything on the Internet.
 - b. Neither application should allow any executable code or scripts to be accepted from the Internet and run.
5. If one uses wireless networking at home, the wireless access point must be placed outside the home firewall, rather than inside. Unfortunately, all current bundled firewall/access point products place the access point inside the firewall. Therefore, if one wants network security and wireless networking, and chooses a bundled product, then one must install a personal firewall on every machine in the house and allow no incoming connections on any of them.
6. For each operating system, there are numerous settings that must be made properly to maximize security. The documents describing such settings run to dozens of pages and need to be produced for each different home operating system.

These well-known security measures are both inadequate and burdensome. They are inadequate because any attack code that manages to penetrate a computer on the home network has free run within that computer. Solving this problem requires new operating system architectures—extremely long-term work. They

are burdensome because with these measures in place, a computer user cannot view many modern Web pages because they require JavaScript; cannot read incoming e-mail transmitted in Hypertext Markup Language (HTML) so that the formatting will be as the sender intended; and cannot offer any Web services to friends out on the Internet.

There is a great deal of work yet to do before we have a good solution for the case of the single home computer connected to the Internet. Meanwhile, we as an industry are actively enhancing the home network. Few people today have real networks at home. Rather they have a single computer with a network connection, either dial-up or broadband. In the future, we anticipate home networks with hundreds of nodes. This future home network brings with it additional security problems that are not addressed by the products available today to secure the home computer and not completely addressed by projected modifications to operating systems that are needed to isolate hostile code from valuable resources within the home computer. This paper deals with those additional issues.

ELEMENTS OF SECURITY

It is a popular misconception that “security” is synonymous with “encryption.” In many cases, confidentiality via encryption is the least important element of a security solution. Network security involves a number of different elements:

1. data origin authentication
2. command authorization
3. message integrity protection
4. message replay prevention
5. data confidentiality
6. key distribution
7. trust versus trustworthiness

Data Origin Authentication

Authentication is often tied in modern systems to integrity protection. To authenticate a message, one needs to establish that it came from a particular source. This can be established by physical point-to-point wiring, but can also be established by the use of cryptography, in which the sender of the message has a secret value and uses that secret value plus the message to compute a check value. The receiver/verifier checks the message origin (and integrity) by verifying that the check value could only have been produced by an entity in possession of the secret value. If public-key methods,

which are known as **digital signatures**, are used, then only the sender needs a copy of that secret value in order to get maximum security. If symmetric cryptography, via what is called a Message Authentication Code (MAC), is used, then the receiver also needs a copy of the secret value. Because there are two or more copies of that value in the system when we use a MAC, there is more opportunity for it to be compromised and therefore it is less secure. However, we still use MACs because symmetric methods are typically much faster than public-key methods. A hybrid scheme is often used, in which public-key methods are used to establish symmetric keys that are used for a short period of time.

Command Authorization

Establishing who sent a message, by authentication, is essential, but it is not enough. For example, there might be an incoming message commanding a home alarm system to turn itself off or a message to a home PC asking for a copy of a sensitive file to be sent to the requester.

An incoming message might be characterized as “Hi. I’m X. Do Y for me.” Authentication verifies that the sender was X. Command authorization establishes whether X is allowed to do Y. Until you have established both authentication and authorization, you cannot make a security decision (namely, whether or not to do Y in response to this message).

Message Integrity Protection

It is essential to establish the integrity of incoming messages. This process is usually tied to authentication.

If the attacker could get a copy of a message saying “Hi, I’m X, do Y” and turn it into a message saying “Hi, I’m X, do Z,” then if that new message passed the authentication verification process, the attacker could achieve a result that the legitimate parties did not desire. Normal authentication methods (digital signatures or MACs) include the entire message in the authentication and verification computation, so that any change to the body of the message would invalidate the authentication.

Message Replay Prevention

The attacker might capture a copy of a legitimate message, “Hi, I’m X Turn off the home alarm system.” That attacker could then re-use that message without any modification to it at all, except that it was sent at a time of the attacker’s choosing. This is called a “replay attack.” To prevent it, one must design network protocols that have unique, verifiable information (often

called “freshness data”) included among the data authenticated and verified in each message. This freshness data is often a sequence number or a time value. However, for home network use, especially when there are VCRs blinking 12:00 because the homeowner chooses not to set the clock, it is preferable not to rely on clock values being correct.

Data Confidentiality

Confidentiality could be achieved by dedicated, private network wiring but cryptographically it is achieved by encrypting the contents of the message. As with authentication, there are both symmetric- and public-key methods for doing this. In public-key systems, the receiver has the secret (called a private key); therefore, only the receiver is capable of reading a message encrypted for its key. In symmetric-key methods, the sender also needs a copy of the secret (the symmetric key) and as a result it is less secure. As with authentication, a hybrid method is often used: public-key methods are used to establish symmetric keys that are used for a short period of time or for a single message.

Key Distribution

Both authentication and confidentiality require the two communicating parties to have certain cryptographic keys. If public-key methods are used, the key distribution problem is a little simpler, but it is not trivial. It must be designed very carefully. Flaws or shortcuts in key distribution can completely invalidate the security benefit of the mechanism used.

Unfortunately for home networking, key distribution is considered an onerous task, and shortcuts are often employed to save the homeowner from having to do “geeky” things. So, for example, wireless network devices often come with built-in default keys that homeowners are allowed to just use. Use of such keys makes the security mechanism worthless, but the 802.11 devices don’t know they are using worthless keys, so they spend the same amount of processing time (reducing network bandwidth) as they would with valid keys. Similarly, firewalls often control access by password and come with a default password (e.g., “admin”). Users who leave that password unchanged have completely invalidated the security mechanism.

How keys are distributed varies from one security tool to another and is discussed in more detail in a later section.

Trust Versus Trustworthiness

People sometimes use the words “trusted” and “trustworthy” as if they were synonyms. In fact, they are practically antonyms.

If a thing is trustworthy, then if you trust it you are not exposing yourself to risk. However, a thing is often called “trusted” not because it is trustworthy but because you are forced to trust it. In that case, you are exposed to risk. As a rule of thumb, it is good to have trustworthy things and bad to be required to trust things.

Unfortunately, we have no sure means of establishing trustworthiness when it comes to security. Therefore, it is standard practice to assume an entity is untrustworthy until proved otherwise. This is counter to standard social practice and calls for care on the part of the product designer. A homeowner should not have to rely on trust when it comes to friends or family using devices within a home. Rather, a product needs to be designed where rights can easily be granted to friends, the minimum rights necessary to do the job. Total access should generally not be granted to anyone except the homeowner regardless of how trustworthy the person is.

HOME NETWORK SECURITY REQUIREMENTS

The requirements for security in a home network depend on how “home” is defined. It also depends on what is envisioned as the network within that home.

If the network is just a link from a cable modem to a single PC, then one length of network cable would accomplish all the network security that the homeowner needs. However, we think ahead to a time in the not-too-distant future when a home contains dozens, if not hundreds of networked devices, some belonging to the entire household and some belonging to individuals within the home.

We summarize the security definitions of the previous section in two categories: authorization and confidentiality. For each device in the home network, we need to concern ourselves with two questions:

1. *Authorization*: Which things are authorized to do what actions or access what data on each device?
2. *Confidentiality*: Which things are allowed to read the messages being transferred to a given device from somewhere else?

The “things” referred to here could be networked devices or could be applications on a networked computer being operated by a particular person. Universal Plug and Play (UPnP^{*}) calls these things

* Other brands and names are the property of their respective owners.

“Control Points” (CP). These CPs might all be within the home, but they might also be remote from the home, connecting into the home from the Internet.

Let us look at the definition of “home” more carefully, since people often use radically different definitions for the term without examining those definitions.

Single-Person Homes

The most basic home environment is a dwelling with only one person living in it. All the devices within the home belong to that one person. It is easy to provide a secure home network in such a home, assuming it is not connected to the Internet. Any device within the home can do anything with any other device within the home. One can, for example, use only a wired network and have no other security. If such a home network uses wireless networking, one can make sure that link encryption is used to enforce the policy that only home network devices are allowed to connect to wireless access points within the home.

This most basic home is of little interest, but it is the model that many security designers assume.

When the home network is connected to the Internet, the domain under consideration is no longer the home. It has many people, some to be kept out at all costs and some to be allowed access, but only to carefully selected resources.

Couples With Small Children

The task of securing the network in the home of a couple with small children might be as easy as that of a single person, provided the two adults agree on the security policy.

Families With Teenagers

Life becomes more complex with teenagers. Most teenagers are trying to establish some degree of independence. This might include ownership of personal networked devices and probably would include inviting friends into the house. What if those friends want to plug their own networked components into the home network? The establishment then of a security policy becomes much more complex than it was in the single person’s household.

How much autonomy does the teenage child need? How much autonomy must the child’s guests be allowed? How much does the head of the household have to trust either the child or the child’s friends?

Adult Guests and Roommates

Adult guests and roommates are presumably more trustworthy than the guests of teenage children, but by the *Principle of Least Privilege* (that no person should be granted more access than he or she needs to do his or her job), the same questions apply to adults as to teens.

SECURITY DOMAINS AND POLICY

For the purposes of this paper, let us define a security domain as a set of objects that are allowed to interact with each other. A person is yet another object, according to this definition, although a person is usually represented on the network as an application that has access to a particular private key and can be operated only by a particular person.

A security policy is the specification of how objects in a security domain are allowed to interact.

The objects in these domains are all networked and computerized, for our purposes, but they are not all network components. For example, a networked home alarm system might be in a security domain with one particular control application on the family PC that can only be accessed by one user (let's say the head of the household). Other persons or other applications on that PC would not be allowed access. Another example might be in a single-occupant home where the PC has a directory of financial files that can be accessed by the homeowner *and* by the homeowner's tax accountant (on an office computer, connected by the Internet into the home). In this case, a security domain that includes that specific directory, the accountant's application, and some of the homeowner's applications might need to be defined.

The actual specification of security domains is up to the owner of the resource(s) being protected. What product designers and researchers need to be aware of is that these domains will contain objects that are much finer grained than network nodes, and that a resource owner might define as many security domains as he or she today defines file folders. In other words, some people would define only one while others would define hundreds.

UpnP*, described below, was designed to take this into account. The "object" could be as fine grained as one action performed by one process in one PC running logged in as one particular user, or it could be as large as an entire networked device (e.g., a printer or scanner).

* Other brands and names are the property of their respective owners.

Interacting with these objects are what UPnP calls "Control Points" (CP). The user can define an arbitrary number of security domains in this structure. The user can also define named groups of devices and CPs. In the simplest form, there would be one policy statement: "my Control Points can do everything with my devices." In the most complex form, each pair-wise association would be defined carefully and intentionally.

KEY DISTRIBUTION MECHANISMS

It is not possible to say that one element of a security solution is more important than another, with the implication that you can do just the important parts. Doing 80% of a security solution is like closing 80% of a submarine's hatches and diving. [3]

That said, key distribution is the first and arguably the most important part of a security solution. Included under the term "key distribution" are the following:

1. passwords
2. DES, AES or WEP keys
3. public keys
4. PKI

Passwords

Typed passwords are typically converted by algorithm to cryptographic keys. When they are not converted to keys, they are used for authentication, just as a key is. Therefore, we consider passwords to be in the category of keys. Passwords can be distributed by being set by the manufacturer and printed for the user to read, but they are more secure if the user chooses a password and uses that. A fundamental problem with passwords though is that for security reasons, they should never be written down, but in reality, they are often written down. Most people cannot keep passwords in their memory unless they are very simple. This makes passwords a weak form of security: if they can be memorized, they are probably too simple and so can be guessed; if they are too complex, they are written down and are therefore available to a passerby.

DES, AES or WEP keys

There are symmetric encryption algorithms, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), and protocols using symmetric algorithms, such as WEP (Wired Equivalent Privacy). The keys for these algorithms and protocols are like passwords, in that both ends of a communication need to know the key. These are typically expressed in HEX digits and have the advantage that they can carry more

entropy (information) than a typical password, but have the disadvantage that they are not memorable and must therefore be written down. That makes them potentially available to someone other than the user although not to attackers on the Internet.

Public Keys

Public-Key (PK) cryptography differs from symmetric-key cryptography in that one encrypts with a different key from the one used to decrypt. It is also a characteristic of PK systems that one key, called the private key, can easily be used to generate the other key, called the public key, but the reverse is not true. One cannot easily use the public key to generate the private key. This allows the public key to be published.

When one encrypts with the private key, one gets what is called a "signature." Anyone with access to the public key can verify that this encrypted quantity was encrypted by the private key that corresponds to the public key that verifies the message.

When one encrypts with a public key, one gets confidentiality. Only the holder of the private key can decrypt the message thus encrypted.

For key exchange, PK systems have an advantage in that the public key can be transmitted without any need for privacy. In particular, it can be transmitted over the network. A public key can also be stored wherever it is needed without any efforts to keep it secret, although it must be protected from being replaced with an attacker's public key.

Although a public key can be transmitted without special protection, the machine receiving it needs to decide whether to use that key for a particular purpose. It is that decision, entirely within the receiving machine, that constitutes the security of a public-key distribution mechanism. The keys may flow freely, but there is a security decision to be made in any acceptance of such a key.

Public keys are simpler for a user than are passwords, DES, or WEP keys with which the user needs to enter the actual password or key over a secure channel. Typically this is done by typing. Public keys can be sent over the network and the user need only say "yes" or "no" when the machine that received the key asks if it should accept that offered key. The user can make that determination by comparing keys, without having to type any values. Typically the user will compare some function of the keys, such as a SHA-1 (Secure Hash Algorithm revision 1) hash.

PKI

A traditional Public-Key Infrastructure (PKI) is a mapping from names to public keys, with that mapping created by some trusted third party (usually called a Certification Authority or CA). It sounds good at first, but turns out to have severe problems.

Humans use names. They prefer to deal with them, especially over nonsensical things like keys or hash values. The issue is where those names come from.

With a traditional PKI, the CA must come up with the name to bind to the device's public key and must do that without knowing anything about the person who will eventually look at that name and try to make sense of it. These names must also be unique among all keys being certified. So, for example, a CA might create a name like *Acme MP3 server, model 5489023-M, serial number 20020115-598003*. The user of the name, on the other hand, needs only to distinguish this device from other devices the user owns or otherwise has to deal with, so for the user a name like *MP3s* might make more sense. If the user has two MP3 servers, the second one might be named, by a CA, *Acme MP3 server, model 5489023-M, serial number 20020115-598083*. The user, however, if selecting his or her own name for the device, might call it *bedroom MP3*.

The UPnP* Security Key Management Choice

For UPnP Security, we looked at the methods of key distribution and decided to use public keys and also to name keys personally. In other words, a user would acquire a new device and learn from that device the SHA-1 hash of its public key. That public-key hash is reported to an application the user runs, called the Security Console, and the user gets to compare what was reported over the network to what was learned from the new device (e.g., printed on a card shipped with the device). After a satisfactory comparison, the user then names the key with some name meaningful to the user. From then on, the user refers to the device by that name.

AUTHORIZATION MECHANISMS

Once a key for a given device or component or user has been learned, that entity can be authenticated, but a security decision cannot be made based only on authentication. A device must know what each authenticated entity is allowed to do. Devices cannot be

* Other brands and names are the property of their respective owners.

manufactured with that knowledge built in, so it is the job of the device owner to implant that information.

There are many mechanisms available for this, but the three predominant ones are an Access Control List (ACL), an Authorization Server, and an Authorization Certificate.

Access Control List (ACL)

An ACL is a protected table residing in memory in the same device as the resource whose access is being protected. It is an array of entries, and each entry contains the following:

1. *subject*: an identifier of the entity being granted access
2. *authorization*: an indicator of the rights being granted that subject
3. *delegation*: a flag, indicating whether the subject may further delegate these rights
4. *validity*: optional conditions on validity of the entry, such as a “not-after” date and time

Some ACL entries contain fewer than all four of these fields, but these are enough to cover any home network authorization decision we have encountered.

A device can control access by an ACL alone. This makes programming easier and also allows an access entry to be deleted with ease, assuming one can access the device holding the ACL. It has the disadvantage of requiring a great deal of ACL editing if there are a large number of ACLs or a large number of subjects. It also could require a large amount of ACL storage. Since ACLs must survive power failures, this memory must be non-volatile.

For example, a traditional time-sharing file system ACL would contain a username (or group name) as the subject and some set of file permissions as the authorization (e.g., {read, append}). It would typically not allow delegation or have expiration dates.

The application SSH (Secure Shell) uses a file `.ssh/authorized_keys` which is an ACL whose entries contain only subject entries. Each subject is a public key. The authorizations are all the same (the ability to log in on that account and to do SCP (Secure Copy) commands to it). There is no delegation or validity interval.

In Universal Plug and Play (UPnP[®]) Security, an ACL can have all four fields. The subject is either the hash of a public key, a name of a group of keys, or the reserved element “<any/>.” The authorization is an XML (Extensible Markup Language) element with sub-elements listing individual permissions being granted. Since the subjects are public keys, the subject is able to delegate rights via authorization certificates, so there is a delegation field, with the default being permission to delegate. Validity fields are available if desired.

Authorization Server

If one has an environment (e.g., in a corporation) that contains a large number of devices all of which need the same ACL and if that ACL is very large, because there would be a large number of subjects, and if network costs are low, then it might make sense to move the ACL from each local machine to a server, often called an *authorization server*. This solution does not apply to the home environment, but products are typically developed for both environments at once, by people trained in the corporate environment, so an authorization server might be considered for home use.

However, this does not eliminate the need for an ACL in each device. When one uses an authorization server, the device would generate a message of the form “May X do Y?” send that message to the server and get back an answer, ‘yes’ or ‘no.’ That message from the server back to the device needs to be secured in all the ways described above under the definition of security. Each reply from the server needs to be protected from modification, replay, or imposture. Therefore, it needs to be authenticated and authorized. Since this is the message from the authorization server, one cannot use an authorization server to authenticate and authorize this message. Therefore, the device needs an ACL listing the authorization server. That ACL, in effect, grants all access rights to the server and allows it to delegate rights to others.

Even though each device needs an ACL, there might still be advantages to using a server. The ACL in each device is very small (one entry) and should rarely have to change.

For home use, however, an authorization server probably makes little sense. It complicates the network and adds cost in an environment where there is likely to be very little duplication of devices and therefore little benefit from the consolidation of ACL entries in one server.

* Other brands and names are the property of their respective owners.

Authorization Certificate

Another way to administer authorization without requiring each device ACL to list each subject and its access rights is to allow delegation by way of authorization certificates [1]. An authorization certificate is a digitally signed ACL entry.

A subject listed in the device ACL might be given the right to delegate some set of permissions. That subject can delegate permissions on to a second subject, where what gets delegated to the second subject is the intersection of the rights granted the first subject and the rights delegated on to the second.

With delegation of rights, the burden of administering security is spread out. One could also spread this out by allowing multiple entities to edit the ACL itself, but in that case, one entity could remove rights added by another entity. The entity empowered to edit the ACL also gets complete access to the device. With delegation by authorization certificate, the entity to whom rights have been delegated does not get total rights to the device, cannot further delegate any more than the rights it has been given to delegate, and cannot remove rights of others.

UPnP Security supports authorization certificates although their implementation is at the discretion of the device manufacturer.

Group Definition Certificates

Another way to spread out the administration of authorization is to have ACL entries (or authorization certificates) that grant rights to named groups.

A name in this context is not just a text string. There is no source of globally unique text string names for arbitrary objects nor is there likely ever to be. DNS (Domain Name System) is a working global name space, but the political attacks mounted on the Internet Corporation for Assigned Names and Numbers (ICANN) shows that even DNS is under siege. However, we need globally unique names to avoid ambiguity that can be exploited by an attacker.

For the purposes of this paper (and for UPnP Security) we define a name to have two fields:

1. hash of a public key
2. text name (as defined by the holder of the private key corresponding to the public key)

The pair is globally unique because the hash of the public key is globally unique.

We allow named groups to be defined in the Simple Distributed Security Infrastructure (SDSI) style [4], by a name definition certificate containing the following:

1. *issuer key hash*: the hash of the public key of the entity defining the name
2. *name*: the text of the name being defined
3. *subject*: the specification of the group member, either the hash of a key or a name of a subgroup
4. *validity*: a possible limitation of the lifetime of this group membership, e.g. via not-after dates

This name definition is then digitally signed by the issuer key and stands for the statement that “the subject is an element (or subgroup) of the specified named group.”

With named groups, one can share the administrative load of granting access, but with more limitation than with authorization certificates. Because the name definition certificate contains no authorization field, every entity in the group gets the same access grant as every other entity in the group, that being the access granted that named group in an ACL entry or authorization certificate.

UPnP Security allows named groups, but their use is at the discretion of the device manufacturer.

SECURITY PRODUCTS

The products available in 2002 tend to support the hardened perimeter model of security. This is appropriate to the most basic concept of home (with only one user and no interactions with the outside world) but not to the more complex forms of home environment.

These products also tend to have been designed based on requirements of industry rather than of the home, making their administration difficult and sometimes assuming the existence of both physical security and a group of on-call support professionals.

Universal Plug and Play (UPnP^{*}) Security, described below, is a new standard designed for home use, but is too new to have any products for sale as of the fall of 2002.

Firewalls/Gateways

An Internet gateway or firewall secures an internal network from the Internet, to the extent that it blocks

* Other brands and names are the property of their respective owners.

unsolicited traffic from the outside. As long as there is a single security domain inside the home (a single-person home or a couple with small children, for example), the home can be secured by a single firewall. However, if there is more than one security domain inside the home (e.g., roommates or guests) then a single firewall would not help guard the interests of one internal security domain from other internal nodes. One might create a separate wired network for each security domain and give each of those networks its own firewall. However, that solution gets expensive as the number of domains increases.

Even in homes in which there are multiple security domains whose security is defined through mechanisms other than firewalls, one will probably want a firewall to protect the collection of domains from hostile outside entities.

Wireless Security

Wireless networking is becoming popular at home. It relieves the homeowner of the work of running network wires through and within finished walls. It can also reduce the clutter of wires within a room.

However, with this benefit comes a security drawback. By relieving the homeowner of the work of individually running network wires to each device in the home, wireless networking prevents the homeowner from selecting which devices should connect to a given network as might be accomplished by running wires. Instead, with wireless networks, cryptographic keys need to be used to individually choose which devices should be connected to a network. Devices allowed onto a network would be given the key to use that network.

The choice of wide area coverage networking, as with wireless or power-line networking, might also restrict the number of networks the homeowner could define. With individual wires, the homeowner can set up separate networks for only the cost of some hubs and wires. With 802.11, each separate network would require a separate Access Point and separate channels. Since there are fewer than seven 802.11 channels that can operate in the same area without getting in each other's way, this limits the number of networks that can be declared in a small space like a home and implemented by 802.11.

WEP

Wire Equivalent Privacy (WEP) was the original security measure for 802.11. It has been shown to have a flaw in key usage that allows an attacker to recover the key used after eavesdropping on a few thousand messages.

Therefore, for real security, WEP is not useful. It can be an annoyance for a casual attacker, but not for a determined attacker.

802.11i and 802.1x

There is an on-going standardization effort in the IEEE under the titles of 802.11i and 802.1x to define security mechanisms to replace WEP. Although these definitions will presumably be cryptographically correct, they retain the problem of being wire-like (therefore unable to secure things more fine-grained than whole devices) and being limited by channel assignment. For a home network with a single occupant and therefore a single security domain, this might be a good solution. For a more complex home network, security must be achieved in other ways.

VPN

There are various Virtual Private Network (VPN) products that permit one to associate devices together in virtual networks, each created cryptographically. Most modern VPNs use the IPSEC (Internet Protocol Security) protocol.

With this technology, one can individually connect pairs of machines and build arbitrary security domains, provided the elements of those domains all have IP addresses (are full devices).

One potential problem with IPSEC or other VPN solutions is that if you have a network node (e.g., the homeowner's PC) that is in multiple security domains, a device in one domain might be able to link to a device in another domain by routing traffic through that PC. Preventing this linkage requires proper network administration (e.g., routing tables) within the PC.

UNIVERSAL PLUG AND PLAY

Universal Plug and Play (UPnP^{*}) [2] is an industry initiative designed to make home networking easy. It does not include security in the basic protocol. One can secure UPnP networks by wiring, if there is a single home domain and no wireless or power-line networking. However, in more general cases, one will need UPnP Security.

UPnP Security defines a service to be added to each secured device that allows its security to be managed. It also defines a service and control point behavior for an application called a Security Console, which edits the Access Control List (ACL) of a secured UPnP device and controls other security functions of that Device.

* Other brands and names are the property of their respective owners.

Overview

The basic architecture of UPnP V.1 is client-server, with the client called a "Control Point" (CP) and the server called a "Device." There are three protocols used to let components interact with each other:

1. SOAP (Simple Object Access Protocol): for remote procedure calls from a CP to a Device.
2. SSDP (Simple Service Discovery Protocol): for discovery of Devices.
3. GENA (General Event Notification Architecture): for subscribing to event reports and publication of those events.

SOAP carries the bulk of the work and the security of SOAP is described in detail below. In brief, SOAP is secured by allowing only authorized Control Points to invoke any secured action within a Device. This is accomplished by an ACL in each secured Device, each of the entries of which lists a Control Point (CP) unique ID, a name of a group of CPs or the universal group, <any/>, and what that CP or group is allowed to do on that Device.

SSDP is difficult to secure. It is vital for the authorized user to discover other Devices on the home network, but it is desirable that an attacker not be able to take an inventory of the home network's equipment. Therefore, to secure SSDP, the existence of Devices is announced, but they are announced as generic Devices, and are not described in any detail except in response to requests from authorized Control Points. It is still possible for a determined attacker to take an inventory of a home network, even if all traffic was completely encrypted (e.g., via IPSEC), just based on timing and length of messages. Therefore, securing SSDP is considered low priority until there are significant new research results in anonymity protection.

For security of GENA, we define a normal event variable that anyone can read, named "EncryptedEvent" but it contains any event variables that are to be made available only to authorized Control Points. An EncryptedEvent is sent to a Control Point encrypted in a key known only to the Device and that Control Point. Control Points not allowed to see such an event, are not allowed to subscribe to those events, and are not able to see the events by eavesdropping on the network.

Security Console

UPnP Security has defined a combination Device and Control Point called the Security Console (SC). This can be a separate component or part of some other component, but for the sake of definition, it is treated as

separate. Its purpose is to take security ownership of Devices and then to authorize Control Points (or other Security Consoles) to have access to Devices over which the SC has control.

An SC can own a Device, meaning that it has the right to edit that Device's ACL and can do anything on that Device, or it can have been given some subset of rights to the Device and have the privilege to grant all or some of those rights to another SC or CP (by way of an authorization certificate).

The SC also has the job of defining names of individual Control Points and of groups of Control Points.

Discovery and Component Naming

The first security requirement is to discover your own network components. UPnP uses a broadcast protocol, SSDP, for physical discovery of Devices but nothing for discovery of Control Points. Therefore, the SC, acting as a Device, offers an action by which security-capable CPs can announce themselves to the SC. In that way, it learns of local Control Points while it uses SSDP to learn of local Devices.

Selection of one's own components can be done in a variety of ways, and there is room for much creativity here. UPnP Security offers a generic method, using a Security ID for each component. The Security ID is the SHA-1 hash of a component's public key, expressed in BASE-32 (using only upper case letters and six of the ten digits). It looks like a product registration ID:

GM3GK-RTMOI-4GYK2-ZK5FC-WMTRK

This ID is unique among all components in the world. By comparing the Security ID of a physical component to the ID announced to the Security Console, one can determine precisely which Devices are his or hers, even if the local network contains hundreds of Devices of the same kind (e.g., in a college dorm).

One might also select one's own components physically. For example, if the SC was a handheld unit with a physical link to the component or a very short-range radio link, then selection could be physical.

Once a Device is selected, whether physically or via comparison of Security IDs or otherwise, the user needs a better way to refer to the Device. The manufacturer could supply a name for the Device, but we give the user the chance to set his or her own name for the component. For example, where the manufacturer might specify a name like:

Media Store Model 5328-I-71

a user who thinks of this media store only as a place for MP3 files might choose a name like:

tunes

That name needs to be meaningful only to that user, so the choice of name is entirely up to the user. The name is used by the Security Console user interface, while over the network the hash of the public key (from which the Security ID is generated) is used as the component's unique ID. The key hash is also the ID by which a Control Point is known in a certificate or Device ACL.

Security Ownership

A Control Point does not need to be exclusive about which Security Consoles it advertises itself to. It is the beneficiary of grants of authority and all decision making is done by the Security Console in that case.

The situation is reversed for Devices. A Device has the resources (SOAP Actions) to which access must be restricted. The Security Console, by editing the Device's ACL, tells the Device which Control Points to obey. Therefore, the Device needs to be very exclusive in choosing which Security Console to associate with. This process is called "security ownership," in UPnP Security.

By the generic ownership protocol defined by UPnP Security, an SC can take ownership of a Device only if the SC knows the Device's secret password and the Device is not already owned. Once a device is owned, an SC that owns it can grant co-ownership to another SC or revoke it, but more importantly, an SC that owns a Device can completely re-write the Device's ACL (or do any other ACL editing operation).

Authorization and Permissions

What an SC does by editing a Device ACL is grant authorization to a Control Point or some other SC to exercise certain permissions. These permissions are arbitrary names, chosen by the manufacturer, to correspond to SOAP actions within that device. For example, one manufacturer might choose to have permissions with the same names as the actions that Device offers, with each permission allowing a caller to invoke just that one action. More likely, there will be far fewer permissions than actions and their names will be intuitive to the average user.

In addition, UPnP Security allows permissions to be parameterized, if the manufacturer desires that. For example, the demo of a UPnP Secure Media Server, as presented at the February 2002 Intel Developer Forum, had only one permission "Play" but it was parameterized

with one or more file names, so that one could grant permission to play one or more individual MP3 files.

Delegation and Named Groups

With just ACLs, one can control authorization. However, as the complexity of the home network increases (e.g., when it scales to a college dorm or when it includes older teens and the mobile computers of their visiting friends), maintaining an ACL on each device might prove onerous. If a task becomes onerous, it tends not to be done, hence security is weakened.

The task of maintaining very large ACLs can be made more manageable by the use of certificates. UPnP Security uses both name and authorization certificates, as described previously in this paper.

With both kinds of certificates, the ACL ends up smaller. A possibly large part of the detail of what would have been a large ACL is expressed in certificates instead. When those certificates are issued by someone other than the operator of the SC, this reduces the workload of that operator, spreading it out among others. It also allows a measure of autonomy for teenage children or guests.

Delegation for Constrained ACLs

Some Devices might have very constrained local persistent memories. Since an ACL must survive power cycles, it must be held in persistent memory (such as flash) and a large ACL might overflow the Device's memory.

One might use delegation via name or authorization certificates, not to reduce the manual workload of administering authorization but rather to offload the Device's memory. In the most extreme case, one can have an ACL with only one entry, allowing a particular SC to have and delegate all rights on the device. That SC would then issue authorization certificates to express the same thing that might be expressed in a larger ACL. The user interface for this operation would probably look the same as ACL editing, so that the operator remains unaware of the difference.

OTHER PRODUCT LINES

Universal Plug and Play (UPnP[®]) might not be used for all home security. A UPnP interface consists of SOAP messages and some products might not use SOAP as the preferred protocol.

* Other brands and names are the property of their respective owners.

However, one can use UPnP key distribution and authorization mechanisms (or the equivalent) in all of these product lines. It is key distribution and authorization that provide most of the security and those represent all of the user-visible work.

If a new product development can improve on the UPnP key distribution and administration, then that would be beneficial. Meanwhile, the UPnP forum is working very hard to improve on those areas.

New product developers must carefully consider the following:

- the whole range of environments (definitions of “home”)
- the range of security policies a user might want to establish (e.g., matrices of what component may do what with what other component)
- delegation of rights, via named groups or authorization certificates
- the interaction between home devices and those in the global Internet

The body of this paper should give all the details needed to at least make a checklist for that process. Much more detailed discussion of this process can be found at the author’s personal distributed authorization Web page [5].

CONCLUSION

One conclusion of this paper is that with proper security against the insider threats in the home environment, the security of the home network against threats from outside is increased.

Securing the home network is not the easy job some people would like to believe. A home network security policy can be much more complex than a corporate security policy. The homeowner would have to implement via network security policy controls what the corporation implements via door guards.

Most network security thinking to date has assumed that network access is binary: that one would allow access to the network or not. The idea of controlling access to individual components (or parts of those components, such as individual SOAP actions) is relatively new to network security design. While we adjust our product design process, this will produce a period of gradually increasing security and there will be a gradually lessening tension between the desire for ubiquitous computing and connectivity on the one hand and the desire for real security on the other.

ACKNOWLEDGMENTS

I thank Jesse Walker for keeping me up to date on 802.11i and 802.1x and my fellow members of the UPnP Security team, both within Intel and outside, for the collaborative effort that produced the UPnP Security design and implementation. I also thank Vic Lortz for being the driving force in the UPnP Security effort, chairing the Working Committee and bringing me into that work as security architect.

REFERENCES

- [1] Ellison, et al., “SPKI Certificate Theory,” RFC2693.
- [2] Universal Plug and Play Forum, www.upnp.org
- [3] Jesse Walker, private communication.
- [4] Rivest and Lampson, “Simple Distributed Security Infrastructure.”
<http://theory.lcs.mit.edu/~cis/sdsi.html>
- [5] SPKI and related resources.
<http://world.std.com/~cme/html/spki.html>

AUTHOR’S BIOGRAPHY

Carl M. Ellison is a Senior Security Architect in the Network Architecture Lab of the Corporate Technology Group of Intel Corporation. His current research is devoted to distributed, public-key authorization that can be delegated, as in UPnP Security. He also has a special interest in self-organizing networks. His concentration on security has been a side-effect of a more general career focus on distributed and fault tolerant systems. His e-mail address is carl.m.ellison@intel.com.

Copyright © Intel Corporation 2002. This publication was downloaded from <http://developer.intel.com/>.

Legal notices at <http://www.intel.com/sites/corporate/tradmarx.htm>

For further information visit:

developer.intel.com/technology/itj/index.htm