A Secure Service Infrastructure for Interconnecting Future Home Networks based on DPWS and XACML

Andreas Müller, Holger Kinkelin, Sunil Kumar Ghai* and Georg Carle
Chair for Network Architectures and Services
Technische Universität München
http://www.net.in.tum.de
{mueller, kinkelin, ghai, carle}@net.in.tum.de

ABSTRACT

Home networks differ from most other networks since they are usually administrated by inexperienced users. Today, protocols such as Universal Plug and Play (UPnP) support zero-configuration networking and are used for data-sharing and entertainment. However, security mechanisms are neglected and are not integrated into current UPnP devices. This becomes even more of an issue when we think of future interconnected home networks where many users and devices will interact. A possible successor of UPnP, the Devices Profile for Web Services (DPWS), is built upon the standard Web-Services(WS) stack and thus also provides WS-Security. However, the configuration of fine-grained access rights for DPWS actions (e.g. for browsing through a media collection) is not defined. This paper describes how to use DPWS and the security framework XACML as a basis for a secure service infrastructure for future home networks. Templates for policies can be auto-generated and a trust model based on X.509 certificates is used for identifying devices and for the interconnection of multiple home networks.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Network**]: Miscellaneous

General Terms

Design, Management, Security

*Student of the Delhi College of Engineering and an intern at the TUM

The presented work is part of the AutHoNe project which is partly funded by the German Federal Ministry of Education and Research under grant agreement no. 01BN070[2-5]. The project is being carried out as part of the CELTIC initiative within the EUREKA framework.

HomeNets 2010, September 3, 2010, New Delhi, India.

Keywords

DPWS, XACML, plug and play, home networking, security, trust

1. INTRODUCTION

Home networks call for services that follow the plug and play paradigm. This is mainly because such networks are not administrated by experts and users want to use new developments easily.

One of the driving factors for future home networks are multimedia services such as audio/video streaming applications. Today, UPnP [4] allows to automatically discover a media server, browse through the files and stream data to clients.

However, UPnP offers no security mechanisms and as networks grow, protecting the privacy of data and services becomes more important. Thus, future home networks need a solid security infrastructure that on the one hand provides enough security and on the other hand is easy to use and easy to administrate. A cornerstone for such a solution was presented in [8].

As a possible successor of UPnP, the Devices Profile for Web Services (DPWS) [1] is built upon the OASIS Web Services (WS) stack and implements security by default. However, DPWS offers no possibility to define fine-grained policies for controlling access to certain functions only (DPWS actions). For example, browsing subfolders of a media collection cannot be restricted to a user or a user group. Browsing can be either allowed (if the client provides a valid certificate) or denied.

With the growing bandwidth of home internet connections, users may also desire to share services hosted within their home with the outside world. In fact, we expect that future homes will be interconnected and services will be shared directly between homes. Access control becomes even more urgent when home networks are interconnected. Cryptographic keying material and identifiers for homes and their users/services make secure addressing of services and authentication of users/clients possible.

This paper has the following contributions: 1) a Trust Model for home networking 2) the integration of XACML policies for protecting access to certain DPWS actions 3) a proxy approach for interconnecting home networks using DPWS 4) security mechanisms that are needed in order to secure the interconnection of homes.

The paper is structured as follows: First, Sec. 2 shortly introduces the Devices Profile for Web Services (DPWS) and

the eXtensible Access Control Markup Language (XACML). Sec. 3 then presents the considered scenarios.

Our technical approach together with the basic Trust model is described in Sec. 4. After pointing out some security considerations in Sec. 5, Sec. 6 evaluates our approach and presents a reference example. The paper concludes with a survey of related work presented in Sec. 7 and the conclusion given in Sec. 8.

2. RELEVANT TECHNOLOGIES

2.1 The Devices Profile for Web Services

The Devices Profile for Web Services (DPWS) defines a "minimal set of implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained endpoints" [1]. Originally developed by Microsoft as a possible successor for UPnP and integrated into MS Vista, DPWS was approved as an OASIS standard in June 2009.

DPWS terminology distinguishes between two types of services: a hosting service representing a device (or a server) and a hosted service (the actual service provided by the device). Obviously, a hosting service can host multiple hosted services. DPWS not only builds on web services standards such as WSDL, XML Schema, SOAP, WS-Addressing, WS-Eventing and WS-Discovery, it also supports a subset of the security features as defined in WS-Security [10]. This allows to sign multicast discovery messages and to encrypt the actual data exchanged.

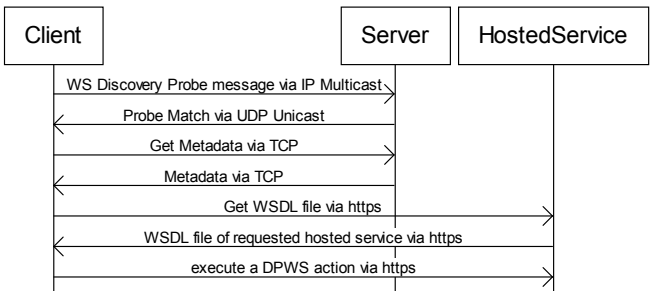

Figure 1: DPWS: Discovery phase

Fig. 1 shows the standard DPWS discovery procedure. First, a client multicasts a probe message looking for a certain type of DPWS device. The server then provides minimal information about a device (DPWS probe match), which the client subsequently uses to request further metadata about hosted services on it. Each hosted service replies with a Web Service Description Language (WSDL) file specifying the datatypes and messages the hosted service understands, as well as the actions implemented by the service.

A hosted service can be protected from an illegitimate access by providing its WSDL file over a HTTPS connection. However, once a client has been authenticated, it is allowed to execute all the actions the service implements. Since in future home networks there may be e.g. guests visiting from several other homes along with the home members, it becomes essential to restrict the access of a service at the action level instead at the service level.

2.2 The eXtensible Access Control Markup Language

The *eXtensible Access Control Markup Language (XACML)* is a language for describing authorization and privacy policies and was standardized by the OASIS Consortium in 2005 [7].

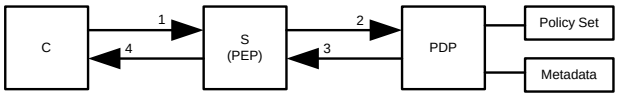

Figure 2: Main XACML Entities

The XACML architecture consists of three main entities (see Fig. 2): a client application C that desires to access a service S (the Policy Enforcement Point (PEP)) and a Policy Decision Point (PDP). Upon a service request of C (1), S creates a XACML query that contains the ID of C (*Subject*), the ID of the *Resource* and the *Action* (e.g. read or write) and sends this query to the PDP (2). The PDP evaluates the query using a predefined Policy Set and additional meta-information and sends the decision back to S (3). The answer (permit or deny) is then dependent on the PDP's decision (4).

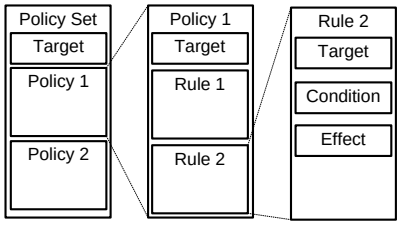

Figure 3: XACML Policy Set

A XACML Policy Set consists of at least one policy. The PDP first evaluates which policy to use by matching the *Target*-Information (Subject, Ressource and Action) of the policies. After finding the right policy the PDP evaluates, which rule to use. A rule specifies if a request should be allowed or denied under which conditions.

3. SCENARIOS

This paper focuses on two scenarios. First, we consider a home network that is equipped with a number of DPWS servers. Members of the home should be able to discover the hosted services and perform the DPWS actions. With standard DPWS, service access (to all actions) is granted if the requester provides a valid certificate. Our solution allows the definition of fine-grained policies to restrict the access to certain actions.

The second scenario covers the access to DPWS services from outside the home network. This requires the cooperation of multiple homes and a security infrastructure to restrict access not only to certain actions, but also to the discovery mechanism of DPWS. Therefore, a secure proxy has to be developed that allows the tunneling of multicast discovery messages across the internet.

4. APPROACH

This section presents our approach for a secure home service infrastructure based on DPWS and XACML. After introducing the entities for our scenarios, a trust model for security in home networks is presented. We then describe what has to be changed if XACML is applied to DPWS in one home while the following paragraphs present the proxy approach and how to secure the proxy using XACML policies. Fig. 4 shows the main entities that are involved in the considered scenarios.

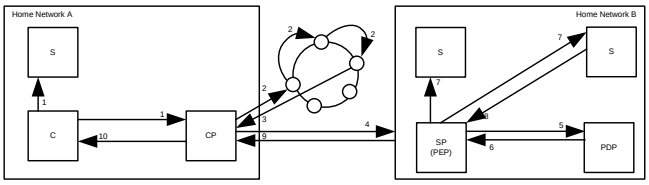

Figure 4: Entities in our scenarios

In the first step a DPWS client C sends a probe multicast message to its own network searching for available DPWS servers. The client proxy (CP) responsible for the client's home receives the messages and looks up the destination IP address of the remote proxy using the mechanisms further described in Sec. 4.1 and 4.3. Once the server proxy (SP) receives the probe message, it checks with the local policy decision point (PDP) if there is any policy matching the incoming packet. This means, the SP enforces the policies defined for the home (policy enforcement point (PEP)). Finally the probe message is multicasted into the local home B and the probe match message can be sent back to home A.

4.1 Trust Model

One of the most important things when introducing new technologies to home networks is usability. In [8] we proposed a security framework for home networks, which acts as the cornerstone for our approach in this paper. The basic idea is that each home runs a local CA with self-signed certificates and manages all devices, services and users belonging to that home in an automated way. Users may "register" new devices to the home network, which is equivalent to requesting a X.509 certificate from the home CA. Thus, one home can be seen as one trust domain.

Interactions between homes will most likely occur between home networks of friends, co-workers and family members. Such groups interact in their daily life personally and share a trust relationship. Our idea is to transfer this personal trust relationship via an easy to understand process to a "virtual" trust relationship between the user's home networks.

For establishing this basic trust relationship, we propose that members of the home networks exchange their home certificates (CA certificate of the homes) as they would exchange a business card. The so called *Trust Exchange* can be performed via a Bluetooth or Near Field Communication (NFC) connection over a secure protocol between mobile devices. The trust exchange satisfies three needs: 1) The home certificates are exchanged securely. 2) As the users performing the trust exchange know each other, they are able to identify each other. The trust into the other person and confidence about the persons identity, can be transferred into the exchanged home certificate. 3) After the trust exchange, both home networks are able to verify certificates issued by the remote home CA as they now trust the remote home CA and possess the remote home certificate. Another approach focuses on deriving trust from social networks such as Facebook and is part of the future work.

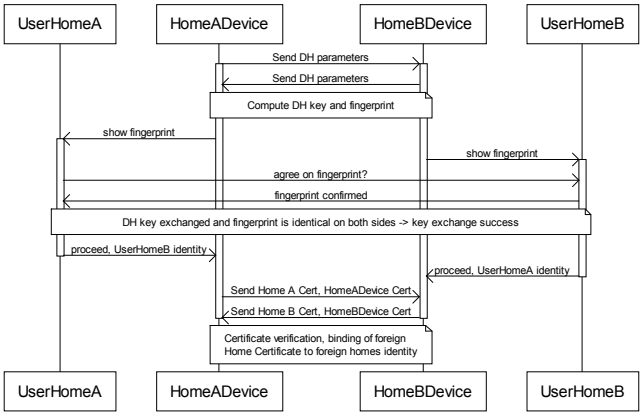

Figure 5: Trust Exchange process (simplified): after an authenticated DH key exchange, certificates are exchanged and bound to the identity of the peer

Cryptographic keys can also be used for addressing home networks. As an alternative to dynamic DNS, a home network can be registered to a P2P network. We propose to use `hash(HomeCAPubKey)` as a globally unique `Home ID`. Retrieving the current IP address of a home is then as easy as quering the DHT for a specific value.

Entities inside a home network have hierarchical, globally unique IDs that consist of the hashed public key taken from their certificate concatenated with the Home ID, for example `localEntityID.HomeID`, and can be reached via their home gateway.

4.2 Equipping DPWS with XACML

After the discovery process the client may eventually want to access a DPWS action as defined in the WSDL file provided over a https connection. In order to restrict the access to certain actions, the DPWS hosting service is extended by PEP functionality. This means, whenever a client asks for a connection to an action, the service not only verifies the provided certificate, but also asks the PDP in the network if the client is allowed to access the action. Thus, once the certificate has been verified, the hosting service extracts the public key from the certificate and creates the client ID by calculating `hash(ClientPubKey)`. The globally unique ID `ClientID.HomeID` is now mapped to the subject field of the XACML request. The resource field is the *ID of the hosted service* and the Action field is the *DPWS action* the client is requesting. The PDP evaluates the request with the help of additional metadata (see Sec. 2.2), which are XACML attributes based on the client ID or its home ID, and sends back the response to the PEP. This provides the flexibility to control the access levels for a remote home, as well as for a specific foreign client. Sec. 4.3.3 shows how templates for these policies can be auto-generated for each hosted service.

4.3 DPWS service usage across homes

DPWS, as well as UPnP, is restricted to only one broadcast domain because it uses IP multicast for discovering devices. For DPWS we might also want to allow the discovery of devices across domains, because we are able to restrict the access on the service itself by using certificates.

4.3.1 DPWS Proxy for interconnecting homes

To enable the remote discovery of DPWS devices we implemented a DPWS interconnection proxy that forwards DPWS discovery messages to remote trusted home networks. The (TCP) connection to the service itself is then established directly from the DPWS client to the server. This is because these messages may be encrypted (SSL/TLS) and cannot be handled by the proxy.

Whenever a device queries the network asking for a service (DPWS probe) the DPWS proxy gets this packet, analyzes it and passes it to the remote DPWS proxy, which sends the multicast packet out to its own network. The remote proxy then acts as a client and therefore gets the reply (probe match) back (see Fig. 7). This requires the proxies to maintain a state and to map between the local network and the identifier of the remote proxy. In our implementation the client proxy also maintains a database where users can configure to which remote homes a probe message should be forwarded. For example, when searching for a media server the client proxy may forward this query to all trusted homes. Remote homes are always identified by a globally unique identifier (see Sec. 4.1) and registered to a P2P network. The client proxy then uses a DHT lookup to retrieve the current IP address of the remote home. Afterwards the proxies establish a secure tunnel (e.g. TLS/SSL) and authenticate each other using service certificates issued by the homes CAs (a trust exchange is necessary in order to verify the service certificates). Therefore, all DPWS servers can be sure that requests only come from trusted home networks. For all other security considerations please see Sec. 5.

Finally, if a network uses Network Address Translation (NAT) the proxy also has to replace the private IP addresses and ports used for establishing the direct connection after the discovery process. This can be done by using NAT-Traversal techniques and frameworks such as ANTS [9].

4.3.2 DPWS Firewall

With the proxy described above the system is now able to discover and use services that are located in trusted remote home networks. The remaining questions now are a) how to suppress the forwarding of incoming probe messages and b) how to make sure that a client is only able to list the services it is allowed to discover. In order to be able to use XACML for this purpose, we equip the server proxy (SP) with PEP functionality. Fig. 7 shows the individual steps that are necessary until a direct https connection to the DPWS action URL can be established. After receiving a signed probe message, the SP extracts the public key from it, calculates the clientID (`hash(pubKey).HomeID`) and asks the PDP if the client is allowed to send discovery messages (1) in Fig. 7). Outgoing probe match messages, as well as metadata files contain a list of hosted services, which may be edited if the client is only allowed to discover a subset of them (2+3). Finally (4), the hosted service enforces the policy as described above.

4.3.3 Policy Creation

The latter approach calls for at least two types of policies. One that defines the access to DPWS actions (e.g. which subject is allowed to access an action of the resource) and secondly, policies defining which remote clients are allowed to discover which internal services. Instead of creating these policies manually, we propose the following: Policies for actions can directly be derived from the appropriate WSDL file of the DPWS hosted service (see Fig. 6). The *Resources* field in the Target section of an XACML policy is mapped to the service ID of a service and the *Actions* field will contain multiple XACML actions; one for each DPWS action (port-type) defined in the service WSDL file. The *Subjects* field in the Target section and the *Rules* in the policy are kept blank and can be edited later by the administrator, if and when required. A default *Rule* can be added as per the network configuration to permit/deny all requests.

```
<Policy>
  <Target>
    <Subjects/>
    <Resources>
      <Resource> <AttributeValue>MediaServer</AttributeValue> </Resource>
    </Resources>
    <Actions>
      <AttributeValue>GetDirectoryContent</AttributeValue>
      <AttributeValue>PlayFile</AttributeValue>
      <AttributeValue>Control</AttributeValue>
    </Actions>
  </Target>
  <Rule Effect="Permit">
    <Target/>
    <Condition> <Apply/> </Condition>
  </Rule>
</Policy>
```

Figure 6: Extracts of an auto-generated XACML policy skeleton

5. SECURITY DISCUSSION

Our system aims at enabling authentication and authorization for service discovery and for the access to DPWS actions. When discussing the security properties of our system, we differentiate between a) access within the home network and b) access across home networks.

5.1 Attacker Model

We assume that legitimate clients possess valid certificates signed by the local or a trusted remote home CA and a Trust Exchange was performed (homes that share a trust relationship are able to authenticate each other). We also assume that trusted homes and entities generally behave well and do not cheat by intention. However, attacks by malware from inside the own home network or from a remote home network can not be excluded.

5.2 Access from within the home

When considering a one domain scenario, our system behaves like standard WS-Discovery. If discovery messages are equipped with a signature, the server is able to verify the probes integrity and the identity of the client. If not, a malicious client or malware inside the home network is able to determine that there are services inside the home network, but is not able to use them since they are provided via an https connection. For breaking our access control mechanism, a malicious client would need to obtain the private

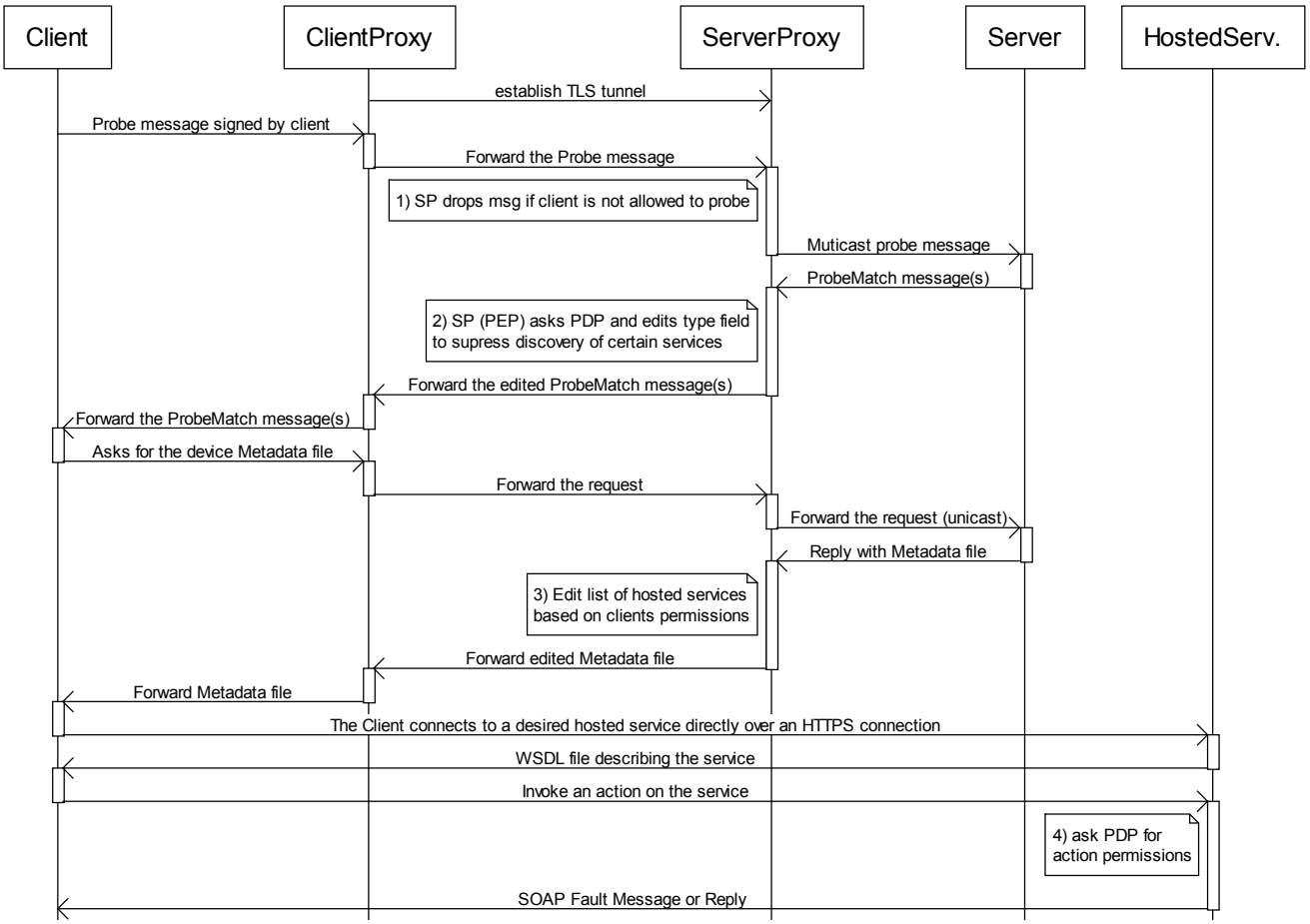

Figure 7: Protecting the discovery and the access of remote DPWS services

key belonging to a valid certificate, e.g. by stealing the private key. Besides the high operating expense of such an attack, private keys could be stored in a Trusted Platform Module (TPM).

5.3 Remote access via proxy

For service discovery and usage between home networks we recommend a higher level of security mechanism: The basic interconnection of two home networks is done via a secure TLS tunnel between the DPWS proxies. As we only forward request originating from trusted home networks, this step limits attacks coming from clients located within trusted networks only and excludes arbitrary attackers located in the public internet.

We suggest that inter domain discovery messages are signed on the message layer using the compact signature format defined in [10]. This enables a home network to authenticate clients from remote homes and determine whether a) the client is allowed to send discovery messages into the home network and b) to filter probe matches according to some policy in order to hide services to clients from remote home networks.

As the compact signature format only contains a signature of the message that will not bind the underlying transport layer to the message. Thus, an attacker in the remote network might eavesdrop a legitimate discovery message and replay it. The only protection against this kind of attack is to log message IDs and discard duplicates. Filtering of duplicate messages can be done locally by the home network, but could lead to additional load (possible DoS attack). As the remote home network is generally trusted, we propose to host this message filter in the remote home. In case an attack takes place, the remote home network can perform countermeasures. Additionally the local home network can filter messages using a short sliding window of message IDs that will limit resource consumption.

6. EVALUATION

To prove that our system works as expected, the DPWS video streaming solutions as described in [8] serves as a reference example for an application that is built upon DPWS. In our opinion, video/audio streaming is one of the most important applications for future home networks and as bandwidth grows, multi domain scenarios are most likely to appear. This means the streaming between multiple home networks should work as automatically and self-configuring as the discovery and streaming within one home.

We then measured the processing time at the proxy between receiving a probe messages from a remote home until forwarding the probe to the local home network. This includes the processing of the messages as well as the XACML query. The DPWS server was not involved, because it sim-

ply implements a second PEP querying the same PDP as the proxy. Table 1 shows the results. The prototype was run on a standard linux machine and the PDP implementation of SUN [1] was used.

# of policies	50	100	500	1000	3000	6000
proc. time in ms	102	205	496	747	1743	3912

Table 1: The processing time in milliseconds dependent on the number of policies

While the processing time of the proxy itself was only approx. 1ms, the XACML lookup is dependent on the number of policies that are maintained at the PDP. We differentiate between policies needed for discovery and policies needed to secure the actions of a service. While every hosted service in the home may get its own policy for discovery, the handling of policies for services is application specific. Furthermore, we propose to run multiple PDP instances in parallel, one for handling discovery messages and one for the applications built on top of DPWS. Thus, the discovery process cannot be blocked by a large number of policies needed for a specific service (e.g. for accessing a media collection).

Another crucial factor is the amount of queries the PDP needs to evaluate. We therefore propose not to authorize local probe messages, but only probe messages coming from outside home networks. The protection against DoS attacks, e.g. by replaying legitimate queries is crucial for the availability of the system. Further work will evaluate the performance of other PDP implementations which claim to perform better than the SUN implementation.

7. RELATED WORK

Several suggestions for securing UPnP [2] and for UPnP remote access [3] have been made and work on enabling new security properties in WS-Security for DPWS exist [6]. XACML as a policy framework has been used for many purposes [11] [5]. However, in the context of home networking and for providing a fine-grained access control to DPWS actions, the use of XACML has not been proposed. This is mainly because providing security features in a way that non-experts can benefit from them is not a trivial task. With our trust model and the decentralized public key infrastructure the described DPWS multi domain scenario is only one example that benefits from the accomplishments.

8. CONCLUSION

Today's home networks offer only few security features and usually run insecure protocols (e.g. UPnP) for applications such as audio/video-streaming. The interconnection of homes, as well as the definition of policies for accessing services are not considered.

This paper introduced a secure service infrastructure for home networks that allows the secure access to DPWS services from within one network as well as from remote home networks. The security framework XACML is used to define fine-grained policies for discovery and service access itself. The whole infrastructure is based on a trust model for home networks that defines how trust between homes and into devices and users can be established without using a central certificate authority. Future work aims at specifying the granularity of policies and at developing forwarding strategies for outgoing probe messages.

9. REFERENCES

[1] D. Driscoll and A. Mensch. DPWS Version 1.1. OASIS Standard Specification, July 2009.

[2] C. Ellison. DeviceSecurity:1 Service Template. http://www.upnp.org/, 2003.

[3] B. N. et.al. UPnP RemoteAccess. http://www.upnp.org/specs/ra/, 2009.

[4] T. U. Forum. Universal Plug and Play. http://www.upnp.org, 2003.

[5] S. Islam and J. W. Atwood. A Policy Framework for Multicast Group Control. IEEE CCNC 2007, Las Vegas, USA, May 2007.

[6] J.-F. Martinez and M. L. et.al. A security architectural approach for DPWS-based devices. CollECTeR Iberoamerica 2008, 2008.

[7] T. Moses. XACML Version 2.0. OASIS Standard Specification, February 2005.

[8] A. Müller, H. Kinkelin, S. Ghai, and G. Carle. An Assisted Device Registration and Service Access System for future Home Networks. IEEE IFIP Wireless Days 2009, Paris, December 2009.

[9] A. Müller, A. Klenk, and G. Carle. ANTS - A Framework for Knowledge based NAT-Traversal. IEEE Globecom 2009, Honolulu, November 2009.

[10] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker. Web Services Security 1.1. OASIS Standard Specification, February 2006.

[11] E. Toktar, E. Jamhour, and C. Maziero. RSVP Policy Control using XACML. IEEE POLICY 2004), June 2004.

[1] http://sunxacml.sourceforge.net