POLITECNICO DI TORINO
Repository ISTITUZIONALE

Toward Cybersecurity Personalization in Smart Homes

Original
Toward Cybersecurity Personalization in Smart Homes / Bringhenti, Daniele; Valenza, Fulvio; Basile, Cataldo. - In: IEEE SECURITY & PRIVACY. - ISSN 1540-7993. - ELETTRONICO. - 20:1(2022), pp. 45-53. [10.1109/MSEC.2021.3117471]

Availability:
This version is available at: 11583/2927212 since: 2023-01-04T16:10:50Z

Publisher:
IEEE

Published
DOI:10.1109/MSEC.2021.3117471

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

Towards Cybersecurity Personalization in Smart Homes

D. Bringhenti, F. Valenza, C. Basile
Politecnico di Torino, Dip. Automatica e Informatica

***Abstract*—Security personalization has become a critical need for smart homes in recent years. Current approaches cannot fully satisfy this requirement of user-centered security. We propose a user-friendly approach for the automatic configuration of home security solutions through policy-based management, minimizing human interventions, and improving security usability.**

NOWADAYS, individuals are exposed to more threats from the outside world, as more risks can undermine cybersecurity in domestic environments. In particular, this larger exposition surface has impacted both the privacy and the wellness of individuals. These considerations are supported by recent investigations about the effects of cyberattacks. According to the most recent Data Breach Investigations Report by Verizon[1], 2020 has seen Social Engineering becoming the most common way to make breaches in networks, and 8% of all the breaches involved humans as unintentional factors.

One of the causes of these worrisome statistics may be that inexpert people have started to use complex devices connected to the Internet. In the last decade, smart homes have become ecosystems where a massive variety of devices coexist, from more traditional personal computers or laptops to Internet-of-Things (loT) or domotics nodes such as locks, sensors, or wearables. The privacy of sensitive data stored on them is threatened by several malicious attacks more than in the past [1]. Another possible cause is the heterogeneity of users. Their need for remote connectivity from home has been progressively increasing since last year when the outbreak of the COVID-19 pandemic urged new ways to live for people of any age — children, middle-aged people, and elderly ones. For example, children may be the victim of episodes of cyber-bullying when using social networks or watch videos with inappropriate content. Instead, older people might have their credit card number stolen, inadvertently activate paid subscriptions, or fall victim to false alarms and fictitious threats. Coping with multiple types of cyberattacks simultaneously is complex, but it is essential so that today's houses are not only smart places where citizens can find more opportunities than in the past but also safe places where they can enjoy those opportunities.

At the moment, several heterogeneous off-the-shelf solutions already exist to challenge the mentioned issues: home gateways, routers, and application programming interfaces (APIs) of devices such as smart TVs and wearables. Examples are TP-Link Router AX1800 and TCL Smart Roku TV 5058435. Despite their low price (e.g., the mentioned router costs less than 100 USD and can be easily installed in home networks without the intervention of a technician), they offer a large number of security services. These services may be locally integrated into the product or remotely accessible through the connection with access

[1] https://enterprise.verizon.com/resources/reports/2021-data-breach-investigations-report.pdf. Accessed: 2021-08-05

 Published by the IEEE Computer Society © 2022 IEEE

points or servers of the Internet Service Provider (ISP), such as Vodafone Secure Net. Besides, they are typically complimentary and could allow offering all-around protection in smart homes. For example, a parental control allows filtering content unsuitable for children, a mail spammer searches for malicious emails to prevent Social Engineering attacks, and a packet filtering firewall offers a thinner granularity for access control. Some of these products may also integrate more complex services, such as identifying intrusions based on attack patterns. Anyhow, it seems that the best solution for protecting our own smart homes is already in the nearest supermarket, and there is no problem that is still worth researching in this environment.

However, people commonly look for and buy off-the-shelf products, such as home gateways or ISP services, mainly to improve the connection quality and solve networking problems related to bandwidth and latency. Even though solutions offer all the security functionalities mentioned before, people typically do not use them [2]. On the one hand, people struggle to understand and endorsing them. Security is already composed of complex concepts by definition, and unfortunately, having to use technical jargon does not help people approach this context. It also happens that the same service is named with different terms in as many solutions, and this does nothing but increasing general confusion. On the other hand, the personalization of these security services is minimal for end users. We have made a non-exhaustive analysis of off-the-shelf products that marketed user-friendliness or simplicity and noted that they also expose security functionalities. However, the behaviour of these functionalities is hardcoded by the manufacturers and can only be turned on or off. For example, parental controls integrated into home-gateways can filter content depending on a specific time of the day, but not always end-users are enabled to change this setting. Facing this limitation, people often decide not to use these services, as the general settings may not match their specific needs. Consequently, solutions that are currently available for enforcing security in home networks are not entirely user-friendly.

One may argue that this decision in the security design is taken to avoid that end-users, inexperienced in security matters, involuntarily open the door to external cyberattacks. However, the problem should be viewed from a different point of view. The challenge should not be understanding how to protect home networks *by limiting* end-users but how to protect them even *when allowing* end-users to customize the security requirements for their smart homes. The direction to pursue should be a *User-Centered Design (UCD)*, where usability goals and user characteristics are given extensive attention during the design of a solution. Specifically, in this context, *User-Centered Security* [3] should be the ultimate objective for home networks: security should be designed so that individuals can understand the main principles behind its services and personalize them in a user-friendly manner.

The achievement of this goal is not trivial in the context of smart homes, however. Manual approaches are not feasible due to the high problem complexity caused by the heterogeneity of available solutions and user needs. Therefore, alternatives must be investigated to improve quality-of-life and personalization of cybersecurity.

In light of these considerations, we have searched for a solution following the *"Keep It Simple, Stupid" (KISS)* principle, which suggests counterbalancing the native intricacy of security problems with the simplicity of their solutions. This direction strictly follows the main ideas behind UCD. To this end, we propose leveraging and adapting a well-known paradigm, already used for business networks, for the security enforcement and personalization in smart home networks: *Policy-Based Management* (PBM) [4]. In particular, the main contributions of this paper are the following: i) the definition of a user-friendly security language that human beings can easily use to define their security requirements in smart homes; ii) the creation of a complete workflow for automatic cybersecurity personalization, that can establish the configuration of security products with minimum external interventions.

Related Work

In literature, user security languages have been proposed to fill the gap between the complexity of security configurations and the simplicity requirement of human users. Originally, standard languages, like Policy Core Informa-

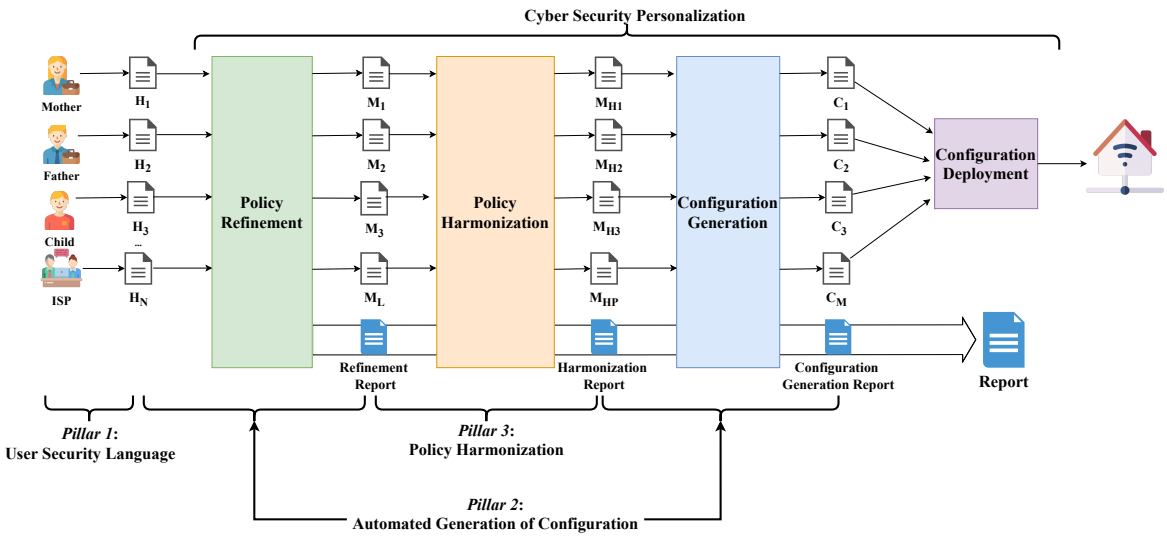

Figure 1. Workflow of automatic cyber security personalization

tion Model (PCIM) proposed in RFC 3060[2] and eXtensible Access Control Markup Language (XACML) proposed by OASIS[3], have been presented in literature. On the basis of them, in more recent years other Access Control Policy (ACP) languages have been studied, such as Ponder [5], SecPAL [6], and other Usage Control Languages [7]. However, all these languages have three main limitations compared to the language that we are proposing:

1) they cannot simply be used by people with no expertise in computer information and mainly target security-savvy people;
2) they do not cover the whole area of network security but exclusively deal with access control;
3) the related research is generally focused on corporate computer networks without focusing on the specificity of home networks.

Similar considerations apply to the other main contribution of our paper, i.e., the application of policy-based management for automating security configuration. This technique lays its foundations on intent-based networking [8], an orchestration principle that aims at minimizing human interventions. In fact, the configuration of network security is automatically generated from high-level intents, also called policies, that describe how the network should behave without specifying all the details. On the one hand, access control and firewall have been deeply studied in the literature, e.g., in the work of Bartal and Bringhenti [9], [10]. On the other hand, little research has been spent on other security controls, such as VPN gateways and intrusion detection systems [11]. Besides, in all cases, policy refinement is not applied to security solutions that are tailored to smart homes.

In light of this literature review, our work is the first combined proposal of a user-friendly security language and automatic security configuration workflow that is specifically designed for smart home networks. At the same time, it also aims to overcome limitations of studies carried out in other areas, e.g., the complexity of usage for security languages and limitation of the configuration computation to specific kinds of security controls.

Approach

We proposed an approach for automatic cybersecurity personalization in smart homes based on Policy-Based Management (PBM). PBM is a management paradigm that aims at abstracting the enforcement of the rules governing the behavior

[2]https://datatracker.ietf.org/doc/html/rfc3060. Accessed: 2021-08-05

[3]http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html. Accessed: 2021-08-05

of systems from their complexity. In this approach, individuals are not required to configure systems manually, but they should just define a set of requirements in the form of security policies. Then, these policies are employed by an automated process for establishing the system configuration, which becomes transparent to end-users. PBM is a solution that has proven to be successful in solving the same problem (i.e., coping with the high complexity of security configuration and customization) in networks of bigger dimensions, such as networks of business companies or universities [11][10]. PBM has become necessary in that context because even expert security administrators could not manually manage huge networks. In smart homes, even though networks are small, people accessing them are less experienced in security; hence, PBM might come in handy.

However, this approach cannot be applied to home networks without a proper adaptation. As previously anticipated, it is crucial to consider the heterogeneity of both security solutions and user categories. Children use home networks to study and play with their friends, parents to work and keep themselves informed, grandparents to remotely interact with relatives who live afar. Each one has multiple and different needs, and their harmonization should be an integral part of the solution. A single security level for a whole smart home is not acceptable, but multiple levels should coexist and not interfere with each other.

The PBM-based approach that we propose to pursue network security personalization in smart homes is illustrated in Figure 1. Three main pillars compose the workflow though which this approach is feasible:

- *user security language*: a user-friendly language allows human beings ease access to the personalization workflow so that expertise in the security fields is not required;
- *automated generation of configuration*: the user-defined policies are refined into an enriched representation, which contains all the information for their enforcement on the security products and services;
- *policy harmonization*: inconsistency among the policies derived from user requirements must be identified and solved to avoid incorrect security behavior.

In the remainder of this paper, we will illustrate each pillar, focusing on its objectives and working mechanisms.

User Security Language

Individuals accessing home networks are commonly non-technically savvy users. They should not learn the needed skills to understand how security services must be configured fully. Instead, they should be allowed to request protection for themselves and their loved ones by expressing security intents, e.g., in natural language, in a way that abstracts the complexity of the security configuration. Then, these security intents should be transposed to a more structured language, named "User Security Language," which represents a link between humans and the automated processes that must actually configure the security. The specification of the security policies is the main operation human beings are required to perform in a PBM-based approach. All the other operations are mostly automated and typically do not involve external interventions.

A language that could match the user needs should have four main characteristics:

- **simplicity**: The language should allow each user to define sophisticated policies intuitively, with statements that can be expressed the most similar as possible to natural language.
- **precision**: The language should accurately represent all the security intents that humans may specify so that it is never ambiguous and can always be used by refinement process to precisely produce the configuration of a corresponding security device or service.
- **flexibility**: The language should support both the heterogeneity of users and security solutions. On one side, it should allow the definition of policies expressing needs for each age (e.g., from studying to working). On the other side, it should not require to cope with the technicalities of the specific security solutions.
- **extensibility**: Every day, new products are manufactured, and new services are available to end-users. If a PBM-based approach does not foresee their possible creation, it fails the objective to achieve User-Centered Security, as it is not future-proofed. Therefore, the

language should support extensions without impacting the structure of the syntax. Users could thus still use the same language just by learning few new words to use.

In light of these considerations, we propose a user-oriented language, called *High-level Security Policy Language* (HSPL) [11], that has the mentioned characteristics. HSPL represents a trade-off between human languages and the traditional approach to configuring products for home security, where users may only enable or disable predefined settings. Therefore, it allows transposing the security intents (which could also be sentences recognized by voice assistants) to structured statements required by the subsequent operation of the approach, that is, policy refinement. The structure of these statements is the following:

$$[< subject >] < action >< object >$$
$$[< (field_{type} = value) > ... < (field_{type} = value) >]$$

In this structure:

- $< subject >$ represents the person for whom the enforcement of the policy is requested (e.g., the child, the grandparent). The subject may be implicit in case there is a single person who accesses the home network. For the same policy, multiple subjects may be specified.
- $< action >$ represents the security operation that must be enforced to fulfill the policy (e.g., block, allow, protect, enable, permit access).
- $< object >$ represents the policy target on which the requested operation must be performed (e.g., email, Internet traffic).
- $< (field_{type} = value) >$ represents an optional condition to characterize the action (e.g., "time = from 9.00 AM to 5.59 PM, GMT", "content type = social networks", "domain = youtube.com"). Multiple conditions may be specified simultaneously to enrich the expressiveness of the policy.

A pair of examples will show how this language can be used for policy specification.

Timed content restriction. Alice is a 10-year-old girl who has been remotely attending school for the last months. Her parents do not want that she accesses social networks during school time. Manual enforcement of this requirement would require configuring multiple services, such as time filtering and web-application firewalls. Instead, with the proposed user security language, her parents can express all the information in a single security intent: "Alice must not access social networks during school time". Then, this intent can be easily transposed in the following HSPL statement:

< Alice >< must not access >< Internet >
[< time = from 10.00 AM to 1.59 PM CET >,
< days = from Monday to Friday >,
< content type = social networks >]

Phishing attenuation. Bob is a 75-year-old man who has been a victim of phishing multiple times. His son, Charles, wants to limit the communications (e.g., through mails, Skype) that his father may establish. Specifically, Bob should be allowed to communicate only with his son and grandchildren. Even though multiple services (e.g., mail system, Voice-over-IP) are involved, again, a single statement is enough to express this security requirement: "Bob can only interact with Charles, David, and Sophie." The corresponding HPSL statement would be the following:

< Bob >< can exclusively start >< communications >
[< source/destination = Charles, David, Sophie >,
< communication means = voice, text messages, emails >]

Besides, HSPL has already proved to be a valid policy specification language; it has been validated in studies [12], [13], where it has been respectively cast into contexts such as smart home and IoT environments.

Interested readers can find more details about this high-level description in past work [11], where the formal specification and examples of the application of the security language are presented.

Automated Generation of Configuration

After the policy specification, our approach envisions an automated process that transforms the policies into an equivalent yet enriched formulation that captures all the required information for the configuration of security products and services to protect smart homes. In literature, this operation is commonly known as policy refinement [14], and in the approach we propose, it works as follows.

Firstly, the policy refinement process identifies the security functionalities required for the

enforcement of the HSPL policies. Not always a one-to-one mapping is possible, as complex policies might require multiple functionalities to be fully enforced. For instance, referring to the *Timed content restriction* example, two security functionalities would be identified: time filtering and web-application filtering. They may be present on the same product (e.g., the home gateway), or at least one of them may be only remotely available. Anyhow, if in the home network of Alice's parents, these functionalities cannot be accessed, this means that the policy cannot be enforced.

Secondly, the configuration of the security products and services that have the identified functionalities is automatically generated. This operation deals with two main issues: i) conciseness of HSPL statements; ii) heterogeneity of security solutions.

Indeed, HSPL simplifies security management for users, but, at the same time, for ease of specification, it must omit details needed for the security configuration. For example, the effective IP addresses that may be used in communications or the network topology are overlooked by this high-level language. Besides, due to the high heterogeneity of solutions, each product requires setting commands characterized by languages with a different syntax, even though they are semantically similar.

To overcome the first issue, this policy refinement process must access external knowledge bases to retrieve information needed for determining the security configuration. Suppose an HSPL statement specifies that a child cannot visit social networks. In that case, there must exist a list of URLs (e.g., a list that is stored in a database and can easily be modified by third-party services or national/international organizations) to be retrieved for computing the configuration of a web application firewall. A similar consideration can be made for the IP addresses of the devices. Traditionally, a mapping between devices and currently assigned IP addresses used to be present in the home gateway, and it used to be employed for computing the rules of a packet filtering firewall. As our vision is to abstract from the specific solutions that individuals may use in their smart homes, our approach aims to embrace a comprehensive view of all the possible databases of IP addresses (e.g., blacklists) and use them to compute the security configurations.

To overcome the second issue, the computed configurations must be expressed with a solution-independent language, i.e., a language that allows the representation of security configurations in a form independent from a specific vendor's implementation. To this end, we propose a language called *Medium-level Security Policy Language* (MSPL). MSPL has a lower level of abstraction than HSPL because an MSPL statement must provide all the required pieces of information, including those retrieved by the mentioned external knowledge bases. At the same time, it is characterized by a generic syntax that abstracts the vendor-specific syntaxes of the different security solutions.

MSPL statements are machine-readable and can be represented as JSON or XML snippets. Referring again to the *Timed content restriction* example, Figure 2 reports an excerpt of the JSON file representing the configurations of the two security functionalities (i.e., time filtering and web application filtering) that are identified to enforce the requested HSPL policy. As seen from this excerpt, the representation is richer than the equivalent HSPL formulation, e.g., it specifies that for each functionality, First Matching Rule (FMR) is adopted as resolution strategy, and the complete list of blocked social networks is present.

After the generation of these MSPL statements, they cannot be immediately enforced on the security solutions that users might access locally in their homes or remotely through their ISP network access points. Therefore, a final translation from MSPL to the syntax of the specific solution is required. However, this operation only consists of a syntax change for the automatic computation of the MSPL configuration. Figure 2 also reports the low-level configuration that would be produced from the corresponding MSPL statement for Squid, a widely used reverse proxy and web-application firewall for Unix-based operating systems. The same information is represented in the two languages (i.e., MSPL and Squid language), simply with a different format.

Then, after this translation, the output needs to be deployed on the security products and services. The ideal case would be, performing deployment

MSPL statement:

```
{
  "subject": "Alice",
  "configurations": [
    {
      "functionality": "timeFiltering",
          "defaultAction": "allow",
      "filteringRules": [
                  {
                   "action": "deny",
                   "condition": {
                          "startTime": "10.00 AM",
                          "endTime": "1.59 PM",
                          "timeZone": "CET",
                          "day": "*"
                         }
                  }
                 ],
         "resolutionStrategy": "FMR"
    },
        {
          "functionality": "L7Filtering",
          "defaultAction": "allow",
          "filteringRules": [
           {
             "action": "deny",
                 "condition": {
                   "url": "facebook.com"
                 }
           },
           {
                 "action": "deny",
                 "condition": {
                 "url": "twitter.com"
                 }
           },
          ...,
         ],
         "resolutionStrategy": "FMR"
    }
  ]
}
```

Squid configuration:

```
acl studying-time time MTWHF 10:00-13:59
acl blacklisted-domains dstdomain www.facebook.com
  www.twitter.com ...
acl all src 0/0
http_access deny studying-time blacklisted-domains
http_access allow all
```

Figure 2. Example of MSPL statement and low-level configuration

through interaction with their APIs. Currently, not all security solutions expose APIs for their configuration, however. Alternative methods (e.g., SSH channels, MQTT) can be used to cope with this limitation, hoping for standardization of these APIs.

At that point, the process automatically produces a report to inform the people who requested the security enforcement about the outcome of their demand. The best result that may be conveyed through this report is that all the policies have been successfully enforced without changing their original specification. Alternatively, it may happen that the policies cannot be fully enforced. For example, if a policy requiring parental control cannot be enforced on a Smart TV, it should be enforced on each application installed. However, it may happen that some applications do not offer this security functionality. In that case, the system reports the effects of the proposed partial enforcement and lists the modified policies. Consequently, the user knows that some applications are still unsafe and may decide to uninstall them or specify a more restrictive policy (e.g., any Internet access from the Smart TV is prohibited).

The proposed process for the automatic generation of security configuration has been originally validated in the context of the EC-funded Project SECURED[4]. The same concepts have been reused in the ANASTACIA Project[5], where an entirely different consortium decided to adopt and customize the SECURED approach. Full details about the use cases and approach validation can be found in the Git repositories of the two projects.

Policy Harmonization

Traditionally, for each product or service, a single security policy used to be defined for all family members. Even when some devices could allow a per-user configuration, this operation was performed by a single individual (e.g., a parent), thus limiting the capabilities offered by the devices. Instead, in our vision, the best approach would be to enable each user the possibility of defining their own policies. On the one hand, this solution simplifies the policy specification itself because the task is distributed among multiple people. On the other hand, additional actors, e.g., ISPs, may want to define policies, and they cannot let their customers personally enforce them in their smart home networks. Nonetheless, inconsistencies may emerge when policies written by different users need to be enforced.

Our approach contemplates the definition of personalized network security policies by envisioning an additional operation called *policy harmonization*. As the naming suggests, the objective of this task is to identify all the policies

[4]https://github.com/SECURED-FP7?language=html
[5]https://gitlab.com/anastacia-project?sort=created_asc

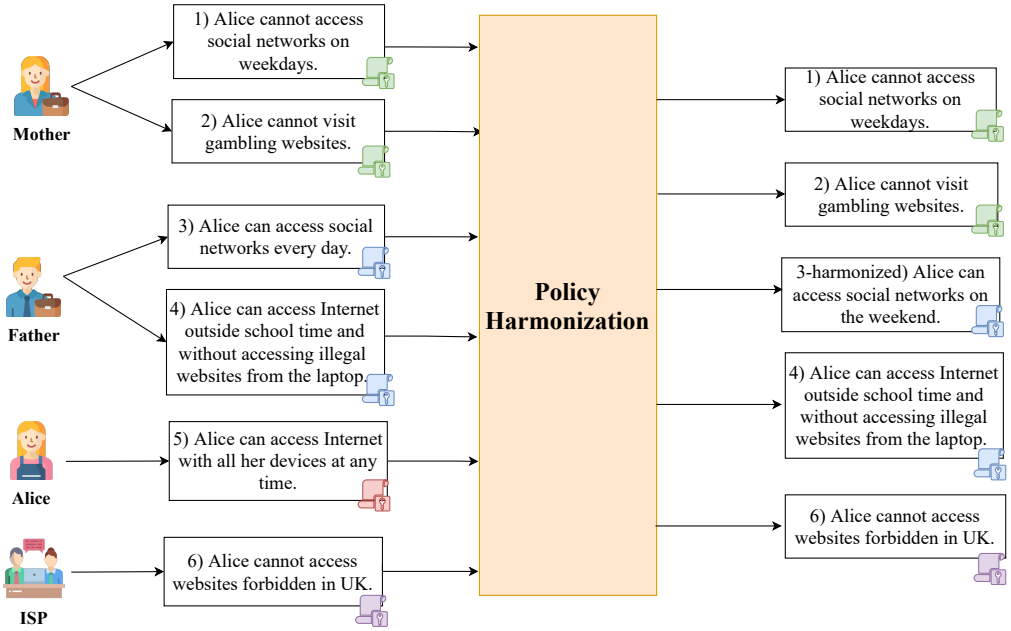

Figure 3. Example of policy harmonization

specified for a specific user and harmonize them. This case is envisioned to consider scenarios where complete enforcement of all the policies is not feasible because contradictions affect them. Contradictions may emerge when users have different security requirements, e.g., when parents can overrule the requests of their children, or when users wrongly define contradicting policies without noticing the errors, e.g., if the parents have to define large policy sets to set up the security for them and their children and do not properly check for inconsistencies.

Policy harmonization is performed according to a reconciliation process, which defines how inconsistencies need to be customized by the experts (e.g., the ISPs or third-party vendors) to allow the users to solve inconsistencies transparently. Policy harmonization is performed in our network security personalization approach after the policy refinement and before generating the low-level configuration. Policies are both specific enough to permit precise identification of contradictions and abstract enough to avoid getting lost by useless details (like devices' syntax). The reconciliation process we have adopted, which is presented below, is based on the feedback of a set of smart home use cases. Besides, it is based on the same formal models that have been successfully employed for the reconciliation of firewall policies in other studies [15].

The proposed reconciliation starts from the MSPL statements, obtained by refinement of the HSPL policies expressed by all family members. HSPL policies are grouped according to their subject (e.g., all the policies for Alice specified by herself, her parents, and the ISP). Then, each group of statements is analyzed to identify the inconsistencies resolved using multiple strategies, which are applied in a fixed order. The complexity of the reconciliation derives from the purpose of hiding to users the complexity of explicitly managing default actions, exception, blacklisting vs. white-listing, and propagating them in different policy sets written by different authors.

Among these strategies, we describe the first ones, which are respectively the *security-first* and *family-role-first* strategies.

The *security-first* strategy implements the security-by-default principle. When two policies for the same subject are contradicting, the most restrictive statement is always enforced. Instead, the other one is modified to be enforced as far as possible, i.e., by "subtracting" the part that the other policy has overruled. This strategy prioritizes the actions that are requested by the policies. For example, "deny," "block,", and "prohibit" are

considered more restrictive than (thus prevail on) "allow" and "permit." To clarify this strategy, let us consider the example shown in Figure 3, where the father has requested for her daughter Alice to access social networks every day, whereas her mother has explicitly prohibited this access type for all the children during weekdays. Policy harmonization firstly groups these policies and identifies a partial overlapping. The latter policy is the most restrictive and is thus preserved in the harmonized set without changes. Instead, the former is modified to access social networks is granted to Alice only on Saturday and Sunday.

According to the *family-role-first* strategy, in case of inconsistencies between policies specified by different users, the policy requested by the person with a higher family role in the home is enforced (e.g., parents can overrule their children's requests) while the other is refused or partially modified. This strategy has a lower priority than the *security-first* strategy. Therefore, the process of policy harmonization applies only when an anomaly still persists for similarly conservative policies, i.e., for policies whose action has equal priority in terms of security. For instance, if Alice requests to access the Internet with all her devices, whereas the father decides that Alice can only access the Internet outside the school time and without accessing illegal websites from the laptop, the father policy is enforced.

Finally, the report produced at the end of the automatic configuration also describes the outcome of policy harmonization. Users must be informed if their policies have only been partially enforced and if other ones have overruled them. The overruling policy is specified as well in the report. This information allows them to understand how different needs were conflicting so that individuals living in the same home can discuss the result of the automatic configuration and decide accordingly. Not always this may be possible, however. For example, if the ISP requests the overruling policy, the hidden visibility of this policy would make the report omit it.

Conclusion and Future Work

Nowadays, many limitations for personalizing cybersecurity in smart homes still exist in current solutions, and individuals commonly struggle to use the security features offered by off-the-shelf products and services. Therefore, to overcome these limitations, we have suggested the possibility of employing an automated approach to simplify and customize security configuration in domestic environments, with minimum intervention of humans. This approach lays its foundation on policy-based management and provides users with a language for policy specification that represents a trade-off between human languages and machine-like representations. An operation of policy harmonization is also envisioned to identify and solve possible anomalies in the policy specification, e.g., when two family members define conflicting policies.

As future work, we are further optimizing the implementation of this approach to minimize the number of APIs required to deploy the automatically computed configurations. Besides, we plan to extend the proposed user security language and the whole approach to work in smart homes and other contexts, such as more general IoT networks or Fog/Edge computing environments.

REFERENCES

1. P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on smart homes system using internet-of-things," in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, 2015, pp. 0330–0335. [Online]. Available: https://doi.org/10.1109/ICCPEIC.2015.7259486
2. N. Taha and L. Dahabiyeh, "College students information security awareness: a comparison between smartphones and computers," *Educ. Inf. Technol.*, vol. 26, no. 2, pp. 1721–1736, 2021. [Online]. Available: https://doi.org/10.1007/s10639-020-10330-0
3. M. Lacoste, M. Miettinen, N. Neves, F. M. V. Ramos, M. Vukolic, F. Charmet, R. Yaich, K. Oborzynski, G. Vernekar, and P. Sousa, "User-centric security and dependability in the clouds-of-clouds," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 64–75, 2016. [Online]. Available: https://doi.org/10.1109/MCC.2016.110
4. A. A. Jabal, M. Davari, E. Bertino, C. Makaya, S. Calo, D. Verma, A. Russo, and C. Williams, "Methods and tools for policy analysis," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–35, 2019. [Online]. Available: https://doi.org/10.1145/3295749
5. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in *Policies for Distributed Systems and Networks, International Workshop, POLICY 2001 Bristol, UK, January 29-31,*

2001, Proceedings, ser. Lecture Notes in Computer Science, M. Sloman, J. Lobo, and E. Lupu, Eds., vol. 1995. Springer, 2001, pp. 18–38. [Online]. Available: https://doi.org/10.1007/3-540-44569-2_2

6. M. Y. Becker, C. Fournet, and A. D. Gordon, "Secpal: Design and semantics of a decentralized authorization language," *J. Comput. Secur.*, vol. 18, no. 4, pp. 619–665, 2010. [Online]. Available: https://doi.org/10.3233/JCS-2009-0364
7. A. Lazouski, F. Martinelli, and P. Mori, "Usage control in computer security: A survey," *Comput. Sci. Rev.*, vol. 4, no. 2, pp. 81–99, 2010. [Online]. Available: https://doi.org/10.1016/j.cosrev.2010.02.002
8. E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *91st IEEE Vehicular Technology Conference, VTC Spring 2020, Antwerp, Belgium, May 25-28, 2020*. IEEE, 2020, pp. 1–5. [Online]. Available: https://doi.org/10.1109/VTC2020-Spring48590.2020.9128422
9. Y. Bartal, A. J. Mayer, K. Nissim, and A. Wool, "*Firmato*: A novel firewall management toolkit," *ACM Trans. Comput. Syst.*, vol. 22, no. 4, pp. 381–420, 2004. [Online]. Available: https://doi.org/10.1145/1035582.1035583
10. D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov, "Automated optimal firewall orchestration and configuration in virtualized networks," in *NOMS 2020 - IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, April 20-24, 2020*. IEEE, 2020, pp. 1–7. [Online]. Available: https://doi.org/10.1109/NOMS47738.2020.9110402
11. C. Basile, F. Valenza, A. Lioy, D. R. López, and A. P. Perales, "Adding support for automatic enforcement of security policies in NFV networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 707–720, 2019. [Online]. Available: https://doi.org/10.1109/TNET.2019.2895278
12. D. Montero, M. Yannuzzi, A. L. Shaw, L. Jacquin, A. Pastor, R. Serral-Gracià, A. Lioy, F. Risso, C. Basile, R. Sassu, M. Nemirovsky, F. Ciaccia, M. Georgiades, S. Charalambides, J. Kuusijärvi, and F. Bosco, "Virtualized security at the network edge: a user-centric approach," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 176–186, 2015. [Online]. Available: https://doi.org/10.1109/MCOM.2015.7081092
13. S. Ziegler, A. F. Skarmeta, J. B. Bernabé, E. E. Kim, and S. Bianchi, "ANASTACIA: advanced networked agents for security and trust assessment in CPS iot architectures," in *Global Internet of Things Summit, GIoTS 2017, Geneva, Switzerland, June 6-9, 2017*. IEEE, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GIOTS.2017.8016285
14. A. C. Riekstin, G. C. Januario, B. B. Rodrigues, V. T. Nascimento, T. C. M. de Brito Carvalho, and C. Meirosu, "A survey of policy refinement methods as a support for sustainable networks," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 222–235, 2016. [Online]. Available: https://doi.org/10.1109/COMST.2015.2463811
15. C. Basile, A. Lioy, C. Pitscheider, and S. Zhao, "A formal model of policy reconciliation," in *23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2015, Turku, Finland, March 4-6, 2015*, M. Daneshtalab, M. Aldinucci, V. Leppänen, J. Lilius, and M. Brorsson, Eds. IEEE Computer Society, 2015, pp. 587–594. [Online]. Available: https://doi.org/10.1109/PDP.2015.42

Daniele Bringhenti received the M.Sc. degree in computer engineering from Politecnico di Torino, Italy, in 2019, where he is currently pursuing a Ph.D. degree in control and computer engineering. His research interests include novel networking technologies, automatic orchestration, the configuration of security functions in virtualized networks, and formal network security policies. Email: daniele.bringhenti@polito.it.

Fulvio Valenza received the M.Sc. and Ph.D. degrees in computer engineering from Politecnico di Torino, Turin, Italy, in 2013 and 2017, respectively. His research interests include network security policies. He is currently a Researcher with Politecnico di Torino, where he works on orchestration and management of network security functions in SDN/NFV-based networks. Email: fulvio.valenza@polito.it.

Cataldo Basile received an M.Sc. in 2001 and a Ph.D. in Computer Engineering in 2005 from Politecnico di Torino, where he is currently an assistant professor. His research concerns software security, software attestation, policy-based security management, and general models for detection, resolution, and reconciliation of security policy conflicts. Email: cataldo.basile@polito.it.