UNIVERSIDADE FEDERAL DE SANTA CATARINA

Uso de ferramenta para edição e teste de políticas de gerenciamento de identidade

Gabriel Vanini Guimarães

Florianópolis, 2019

Gabriel Vanini Guimarães

Uso de ferramenta para edição e teste de políticas de gerenciamento de identidade

Trabalho de conclusão de curso de Graduação em Sistemas de Informação.
Orientador: Prof . Dr . Jorge Werner.

_____
Profa. Dra. Carla Merkle Westphall
Coordenadora

Banca Examinadora:

_____
Dr. Jorge Werner
Orientador

_____
Prof. Dr. Carlos Becker Westphall

_____
Doutorando Leandro Loffi

Agradecimentos

Agradeço primeiramente a Deus pela sabedoria e saúde. Agradeço à todos que me encorajaram e ajudaram de alguma maneira a seguir o curso até o fim, aos professores pelo ensinamento. Agradeço de forma especial à professora Carla e Jorge Werner pelas orientações neste trabalho de conclusão de curso.

Resumo

Com o aumento de dados gerados, provedores de serviços disponíveis e novos dispositivos na conectados na internet se faz necessário tratar a privacidade para que as informações dos usuários estejam seguras. Isso posto, o presente trabalho apresentará ferramentas para que a construção de políticas de privacidade seja menos custosa aos donos da informação visto que os grandes interessados em uma política geralmente não detém o conhecimento específico em tecnologias inerentes. Os testes em ferramenta propiciarão mais clareza sobre a prática de expressar as preferências pelo dono da informação sobre a coleta, uso, retenção e divulgação de seus dados pessoais, provendo assim, maior segurança na web.

Com a elaboração de propostas para gerenciamento de políticas de identidade, tem-se um bom material teórico do tema. Porém, se faz necessário avaliar as propostas a fim de ter um maior embasamento do modelo proposto e, com isso, difundir o modelo estudado à comunidade acadêmica e empresas de tecnologia. O presente trabalho utilizou uma tese de doutorado como base para definições de privacidade e executou testes mostrando na prática suas funções.

Palavras-chave

IoT. Computação em nuvem. Federação. Gerenciamento de Identidades. Privacidade.

Lista de abreviaturas e siglas

SPT - Security Policy Tool

ABAC - Attribute Based Access Control

XACML - eXtensible Access Control Markup Language

PDP - Ponto de Decisão de Política

PEP - Pontos de Execução de Políticas

PAP - Pontos de Administração de Políticas

PIP - Ponto de Informação de Políticas.

ITL - Information Technology Laboratory

NIST - National Institute of Standards and Technology

LGPD - Lei Geral de Proteção de Dados

RGPD - Regulamento Geral sobre Proteção de Dados

UML - Unified Modeling Language

Lista de figuras

SUMÁRIO

1. INTRODUÇÃO

Com o aumento cada vez maior de aplicações web e computação em nuvem em geral, se faz necessário realizar controle de acesso às informações e gerenciar o volume de dados que circulam na rede. O gerenciamento de identidade trata como os dados relacionados a pessoas ou empresas que são usados para processar, autenticar e restringir os acessos na web (WERNER, 2017) A proposta neste trabalho é simular a execução de políticas de privacidade para o gerenciamento de identidade, partindo do pressuposto que o usuário deseja manter em sigilo, seguro e com acesso restringido de seus dados pessoais. As políticas auxiliam o usuário, dono da informação, por exemplo, para poder pré-definir quais atributos estarão disponíveis, para qual finalidade eles serão usados, por quanto tempo ou se deseja cifrar os seus atributos pessoais.

As políticas propostas foram pensadas e desenvolvidas para que qualquer usuário de uma cloud, inclusive leigos, consigam personalizar suas preferências de privacidade. Elas seriam apresentadas em uma aplicação na forma de perguntas, como o exemplo: “Você autoriza a disseminação de dados pessoais de acordo com o propósito específico? Qual?” ou “Você autoriza a disseminação de seus dados pessoais? Por quanto tempo?” (WERNER, 2017, p. 128 e 129). A propostas foram baseadas em 5 características essenciais em relação a privacidade dos dados dos usuários:

- *Controle*: quais atributos serão autorizados a acessar dados de algum servidor
- *Retenção*: por quanto tempo os dados estarão acessíveis à um provedor de serviços.
- *Notificação*: informa os usuários quando e de que forma seus dados foram processados
- *Proteção*: criptografia sobre os atributos
- *Granularidade*: regras de disseminação dos dados de forma minimizada, de acordo com o propósito

A simulação das propostas e conceitos apresentados serão feitos em ferramentas que auxiliam na modelagem de políticas de segurança na qual será analisado se o acesso à informação é garantido, quais atributos são usados, se as preferências do usuário no uso da informação está sendo respeitado, se a privacidade está sendo garantida e, se for o caso, apontar falhas de segurança nas propostas. Serão apresentadas as principais ferramentas e posteriormente a escolha da melhor para os testes e simulações.

Segundo Hadar (2018), citado por (Dinev e Hart 2006) e (Ayalon e Toch 2013), “Vários estudos mediram os riscos à privacidade e a tomada de decisões em domínios como comércio eletrônico e redes sociais on-line”. Esses estudos revelam que, na prática, os desenvolvedores estão dispostos a melhorar o nível de privacidade oferecido aos usuários obtendo uma melhor usabilidade do sistema. A necessidade que os conceitos de privacidade e ferramentas para gerenciamento das políticas de privacidade sejam divulgados é grande e o presente trabalho auxiliará em difundi-lo.

1.1 OBJETIVOS

O objetivo geral deste trabalho é analisar e validar as políticas no gerenciamento de identidade , desenvolvendo estudos de casos em diferentes contextos.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- apresentação dos conceitos com a fundamentação teórica.
- pesquisa sobre contextualização da segurança aplicada aos dados dos usuários em redes.
- implementação das políticas de gerenciamento de identidade, validando-as em ferramenta de segurança
- análise dos resultados e apresentação da utilidade do modelo

1.3 ESCOPO DO TRABALHO

Os resultados esperados do documento de TCC é haver uma revisão bibliográfica dos conceitos, estudo do estado da arte, implementação das políticas na ferramenta, apresentação dos resultados obtidos na avaliação das políticas e possíveis trabalhos futuros.

1.4 MOTIVAÇÃO

Geralmente nas organizações o responsável por gerar uma política não é o mesmo interessado nela. Normalmente é o administrador da rede interna que faz a criação e, em muitas vezes, gera uma dificuldade pela falta de conhecimento específico na área a ser beneficiada com a política. O criador da política deve ser um especialista, para poder descrevê-la em linguagem técnica e o que deseja e se uma nova política não conflita com as já geradas. Assim, tem-se o desejo de apresentar ferramentas para gerenciamento de políticas de privacidade em XACML que sejam fáceis e intuitivas para usar de modo que seja acessível à qualquer usuário.

1.5 PROPOSTA DE TRABALHO

Com as novas leis vigentes como a Lei Geral de Proteção de Dados, marco civil da Internet e tantos outros em vários países do mundo, as organizações devem estar de acordo com tais regulamentações sobre especificamente, no contexto do presente trabalho, a privacidade e uso dos dados dos usuários na rede. Pelo exposto, apresenta-se como proposta trazer ao leitor os novos conceitos que norteiam a privacidade bem como as tecnologias envolvidas. Dando seguimento, será apresentado ferramentas para criação e manipulação de políticas de privacidade, demonstrando também na prática conceitos atuais retirados da tese de doutorado de Jorge Werner, da Universidade Federal de Santa Catarina. Com os conceitos implementados na ferramenta, tem-se um material explicativo sobre novas

formas de modelagem, funcionalidades das ferramentas, novas usabilidades de softwares e novas validações possíveis do que se tem criado. Ao fim, espera-se que todo material documentado contribua para padronização das linguagens de privacidade e difunda as ferramentas com fácil implementação.

1.6 METODOLOGIA DE PESQUISA

A princípio, na etapa inicial será feito a leitura de artigos e teses mais recentes sobre o tema, sobretudo o publicado pelo doutor Jorge Werner, UMA ABORDAGEM DE PRIVACIDADE NO GERENCIAMENTO DE IDENTIDADE NA NUVEM a fim de construir a fundamentação teórica e estado da arte para melhor contextualização do tema por parte do leitor. Posteriormente, o download e cadastro no site oficial da ferramenta SPT (Security Policy Tool), implementação das políticas de gerenciamento de identidade na ferramenta baseada em XACML (eXtensible Access Control Markup Language), na qual, serão avaliadas as teses. Concluído a implementação, haverá uma fase de testes, analisando os resultados obtidos. O projeto será executado obedecendo a metodologia explicativa, analisando as interações entre as regras com resultados. A experimentação das políticas poderão dar a relação de causa e efeito, desempenho, nível de segurança e talvez outros fatores.

1.7 ESTRUTURA DO TRABALHO

Este trabalho está dividido em seis capítulos: O Capítulo 2 é o capítulo que contém toda a fundamentação teórica, abordando conceitos sobre políticas de privacidade bem como sua contextualização, a linguagem XACML e sua interação com o modelo de controle de acesso ABAC (Attribute Based Access Control) e apresentação de ferramentas para gerenciamento de políticas de privacidade. O capítulo 3 começa com apresentação das definições do artigo UMA ABORDAGEM DE PRIVACIDADE NO GERENCIAMENTO DE IDENTIDADE NA NUVEM de Jorge Werner. O capítulo 4 começa o desenvolvimento prático do trabalho. Na ferramenta escolhida, foram modeladas as definições de privacidade da tese juntamente com

alguns exemplos. O capítulo 5 apresenta a análise dos resultados. O Capítulo 6 é a conclusão do trabalho como um todo e discussão sobre trabalhos futuros. Por fim temos as referências.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 POLÍTICAS DE PRIVACIDADE

As discussões a respeito das políticas de privacidade em cloud tem tomado grandes proporções visto o grande crescimento dos usuários e dispositivos na web. Atualmente, os usuários na rede estão mais envolvidos na gestão de suas informações e preocupados na segurança dos mesmos. Segundo Anderson (2006, p. 1), **"**há uma preferência dos usuários em definir suas preferências de privacidade dando-lhes, assim, mais confiança e controle sobre seus dados pessoais."

É importante que as políticas de privacidade descreva a maneira como a informação é manuseada, armazenada, usada e também deve permitir que o cliente tenha um controle máximo sobre os dados e processamento (SADKI; BAKKALI, 2015).

Com toda essa necessidade, a tendência é que cada vez mais empresas adotem os requisitos regulatórios atuais, como Sarbanes-Oxley, HIPAA e Diretiva da União Européia sobre Privacidade de Dados para verificar sua conformidade com as políticas de privacidade (ANDERSON, 2006).

As figuras 01 e 02 não mostram a sintaxe exata do XACML mas sim um pseudocódigo para fácil entendimento sobre uma política de privacidade. Observa-se que aos médicos, sujeito "doctor" a leitura e escrita são permitidos aos recursos registros médicos (medical-records). Na figura 02, a regra 3 permite o acesso de leitura ou escrita aos registros médicos para sujeito "doctor" a pacientes em estado crítico.

Figura 1 - política 1

```
<Policy PolicyId = "Policy 1" rule-combining-algorithm="permit-overrides">
            // Doctor Access to Medical Records //
     <Target>
        /* :Attribute-Category    :Attribute ID     :Attribute Value */
                :access-subject      :Role                :doctor
                :access-subject      :Role                :intern
                :resource            :Resource-id         :medical-records
                :action              :Action-id           :read
                :action              :Action-id           :write
      </Target>
```

Fonte: Ferraiolo, 2016

Figura 2 - regra 3

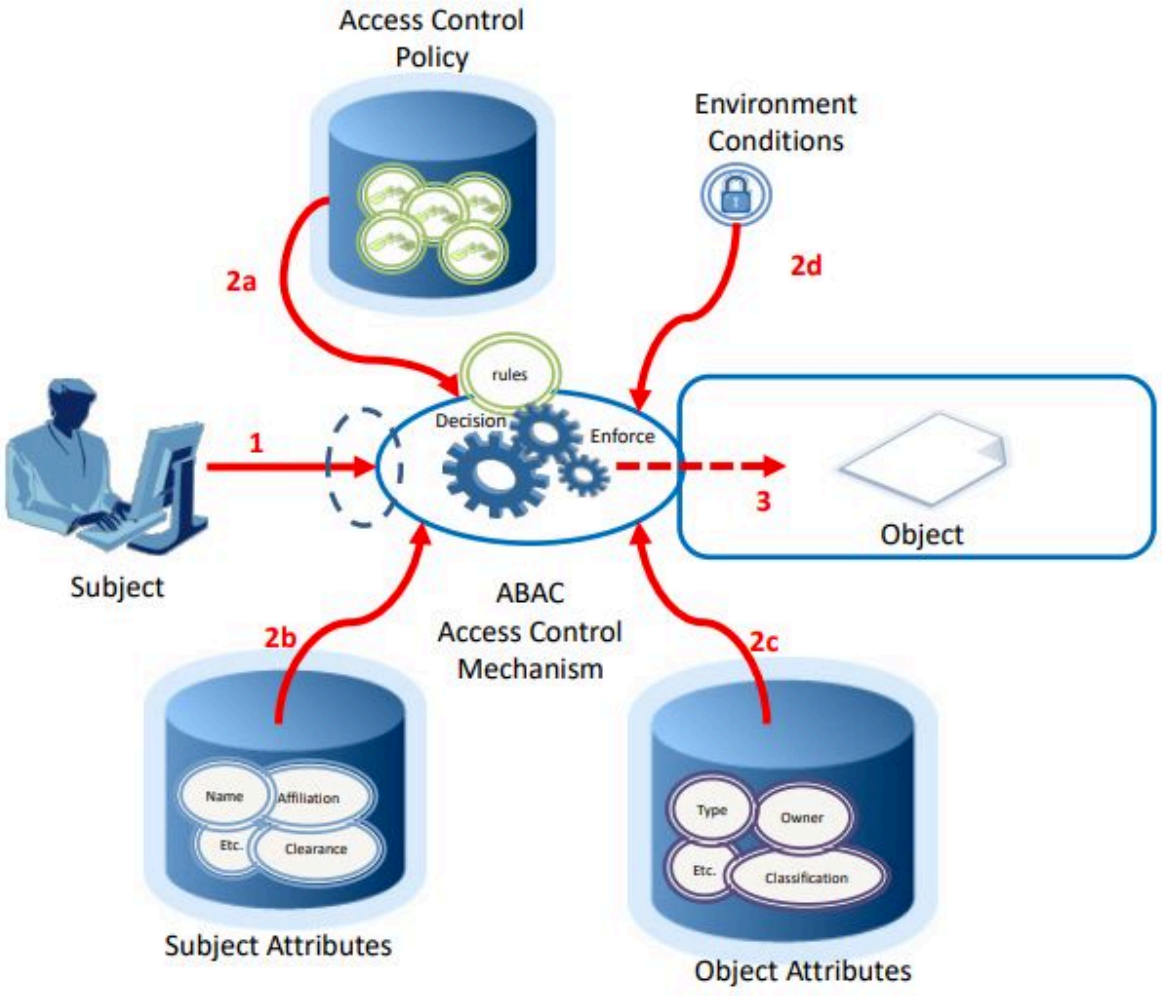
```
      <Rule RuleId = "Rule 3" Effect="Permit">
           <Condition>
        Function: and
          Function: string-equal
            /* :Attribute-Category    :Attribute ID     :Attribute Value */
                    :access-subject      :Role              :doctor
         Function: string-equal
            /* :Attribute-Category    :Attribute ID     :Attribute Value */
                    :resource            :PatientStatus     :critical
           </Condition>
       </Rule>
</Policy>
```

Fonte: Ferraiolo, 2016

2.2 ABAC

Existem várias definições sobre o termo ABAC na literatura. Mas em resumo, ABAC seria um modelo de controle de acesso em que as permissões de execução são concedidas ou negadas a um determinado sujeito com base nos atributos de um objeto.

Figura 3 - Modelo ABAC



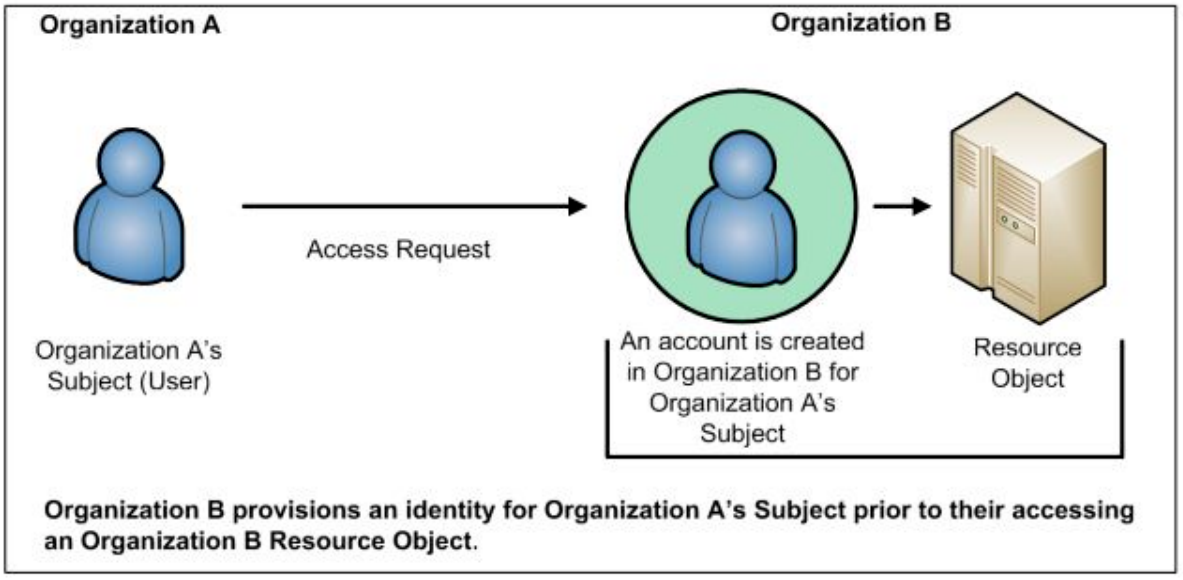
Fonte: (HU, 2014)

“Diante dos modelos de controle de acesso existentes, o ABAC (Controle de Acesso Baseado em Atributo) mostra-se ser o mais completo. Com ele é possível gerar políticas que expressam um conjunto de regras booleanas que pode avaliar muitos atributos diferentes” (HU, 2014, p. 5). Usando a estrutura da linguagem XACML, por exemplo, pode-se manipular regras e políticas, atributos (assunto, objeto, recurso, ação e ambiente) obrigações e conselhos. (HU, 2014, p. 5). Sua arquitetura é composta por políticas: Decisão de Política de Pontos (PDPs), Pontos de Cumprimento de Políticas (PEPs), Pontos de Administração de Políticas (PAPs) e Política de Pontos de Informação (PIPs) para controlar o acesso. (HU, 2014, p. 5). O ABAC é capaz de tomar decisões de controle de acesso a partir do conjunto dos atributos de um objeto, das condições de um ambiente ou conjunto de regras sem

que elas sejam previamente especificadas diretamente a cada sujeito. Diante desses dados é possível criar políticas sem referência direta aos usuários que, podem ser muitos, permitindo gerá-las de forma mais genérica. (HU, 2014, p. 5)

Essa flexibilidade na criação dos controles de acesso é o que dá um grande benefício no uso do modelo ABAC. A figura abaixo retrata um pedido de acesso entre organizações diferentes, que por sua vez possuem representações de identidades e políticas diferentes.

Figura 4 - requisição entre organizações



Fonte: (HU, 2014, p. 6)

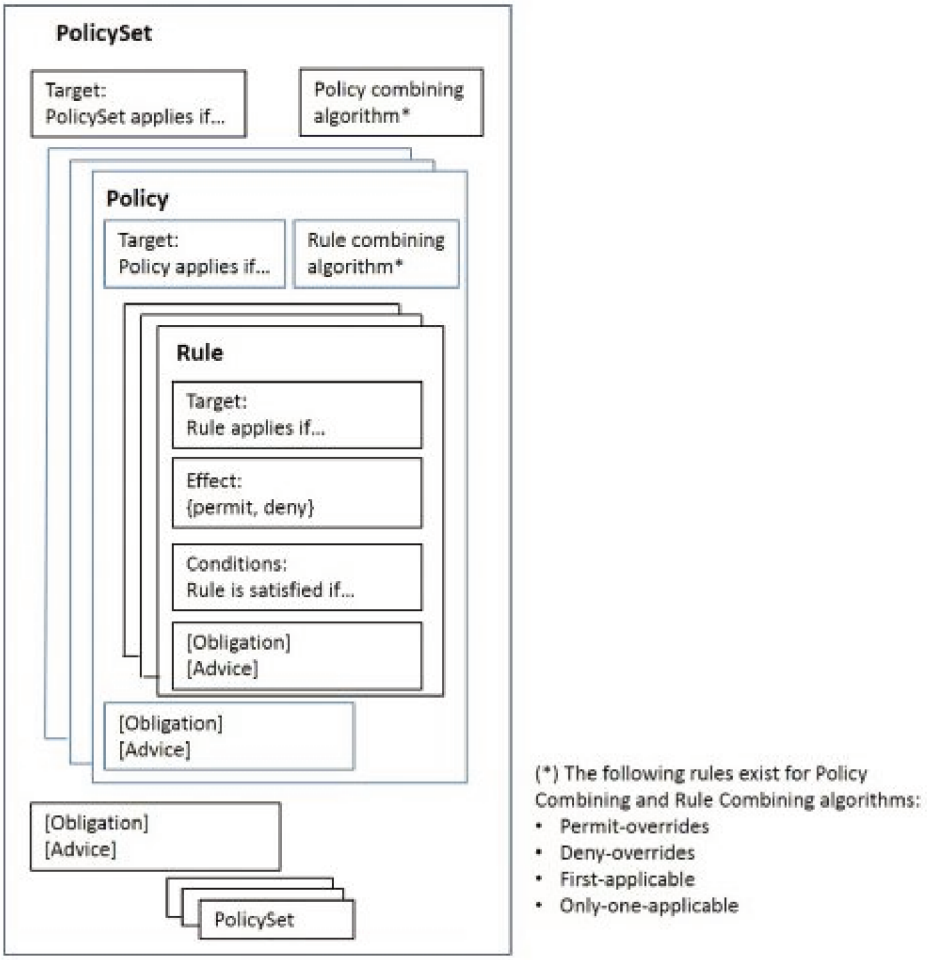
Do modo tradicional, a organização B exigiria que a identidade do solicitante fosse cadastrada e processada do lado de destino, com suas políticas e regras estabelecidas. (HU, 2014, p. 6). Com ABAC é possível tomar decisões de controle de acesso sem conhecimento prévio do objeto pelo sujeito ou conhecimento do assunto pelo proprietário do objeto e evita que a autorização seja atribuída antes da solicitação mas sim de forma dinâmica. (HU, 2014, p. 6).

2.3 XACML

XACML são iniciais formadas de "eXtensible Access Control Markup Language". Ela é uma linguagem de política de controle de acesso que obedece o padrão ABAC e foi desenvolvida pela organização de normas OASIS. Sua primeira versão foi publicada em 2003. XACML define um padrão de especificação que engloba sintaxe e semântica para representar solicitações, políticas, atributos e funções para computar decisões em recurso Ferraiolo (2016, p. 8) e também um modelo para representação das mensagens de requisição e resposta trocadas entre o PEP e o PDP (TOKTAR, 2016). Por se tratar de um padrão aberto, possuem pontos para extensão, possibilitando que um desenvolvedor defina novas funções, tipos de dados, combinações lógicas etc (LIMA, 2008).

Os atributos em XACML também seguem uma padronização. Segundo Lima (2008) toda requisição na linguagem conterá as categorias de atributos do usuário que são sujeito (subject), recurso (resource), ação (action) e ambiente (environment) e são especificados como pares nome-valor, por exemplo Papel = "médico" ou Tempo = “12:11” Ferraiolo (2016, p. 8). A estrutura da linguagem XACML pode ser vista conforme a figura 5:

Figura 5 - estrutura XACML



Fonte: Ferraiolo, 2016

As PolicySets são formadas por políticas e as políticas formadas por regras. As regras podem ser do tipo booleano, “maior que igual”, “menor que” e “string-igual”. Também há possibilidade de fazer cálculos sobre valores de atributos como soma (x, y) menor ou igual a 250 Ferraiolo (2016, p. 10). A resposta de uma requisição têm como resultado uma decisão entre quatro possíveis: permitir, negar, não aplicável (a ação solicitada não é válida) e indeterminado (erro na interpretação ou falha de informação sobre algum atributo).

Como uma Política pode conter várias regras, e um PolicySet pode conter várias políticas ou PolicySets, cada regra, política ou PolicySet pode avaliar decisões diferentes (permitir, negar, não aplicável ou indeterminado). O XACML fornece uma maneira de unificar essas decisões. Esse processo é obtido por meio de uma coleção de algoritmos de combinação. Cada algoritmo representa uma

maneira diferente de combinar múltiplas decisões locais em uma única decisão. Existem vários algoritmos de combinação definidos, segundo Ferraiolo (2016, p. 10):

- Deny-overrides: se qualquer decisão for avaliada como negar, ou se nenhuma decisão for avaliada como permitir, então o resultado é negar. Se todas as decisões forem avaliadas como permitir, o resultado será permitir.
- Permit-overrides: se qualquer decisão for avaliada como permissão, o resultado será permissão, caso contrário o resultado é negar.
- First-applicable: o resultado é o resultado da primeira decisão (permitir, negar ou indeterminado) quando avaliado em sua ordem listada.
- Only-one-applicable: se apenas uma decisão se aplica, o resultado é o resultado da decisão, e se mais de uma decisão se aplica, o resultado é Indeterminado.

Abaixo temos na figura 6 um fragmento de código mais completo em XACML para melhor entendimento da sintaxe. A política refere-se que o sujeito Alice pode acessar seu histórico somente entre 8h e 18h. Das linhas 2 a 6 é feito o início da declaração da política com o elemento Policy. Podemos observar que o atributo xmlns especifica a versão 3 do XACML, o atributo RuleCombiningAlgId especifica o algoritmo de combinação de regras utilizado, no caso o First-Applicable.

Figura 6 - exemplo código XACML

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Policy
3    xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
4    PolicyId="policy1"
5    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
6    Version="1.0">
7    <Description>Alice pode acessar seu histórico somente no horário de trabalho</Description>
8    <Target>
9      <AnyOf>
10       <AllOf>
11         <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
12           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">alice</AttributeValue>
13           <AttributeDesignator
14             AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
15             Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
16             DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/>
17         </Match>
18         <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
19           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
20           <AttributeDesignator
21             AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
22             Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
23             DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="false"/>
24         </Match>
25       </AllOf>
26     </AnyOf>
```

```
27  </Target>
28  <Rule Effect="Permit" RuleId="Permit-Rule">
29    <Target>
30      <AnyOf>
31        <AllOf>
32          <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
33            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">history</AttributeValue>
34            <AttributeDesignator
35              AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
36              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
37              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
38          </Match>
39        </AllOf>
40      </AnyOf>
41    </Target>
42    <Condition>
43      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
44        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than">
45          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
46            <AttributeDesignator
47              AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"
48              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
49              DataType="http://www.w3.org/2001/XMLSchema#time" MustBePresent="true"/>
50          </Apply>
51          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">08:00:00</AttributeValue>
52        </Apply>
53        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than">
54          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
55            <AttributeDesignator
56              AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"
57              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
58              DataType="http://www.w3.org/2001/XMLSchema#time" MustBePresent="true"/>
59          </Apply>
60          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">18:00:00</AttributeValue>
61        </Apply>
62      </Apply>
63    </Condition>
64  </Rule>
65  <Rule Effect="Deny" RuleId="Deny-Rule"/>
66 </Policy>
```

Fonte: Silva, 2018

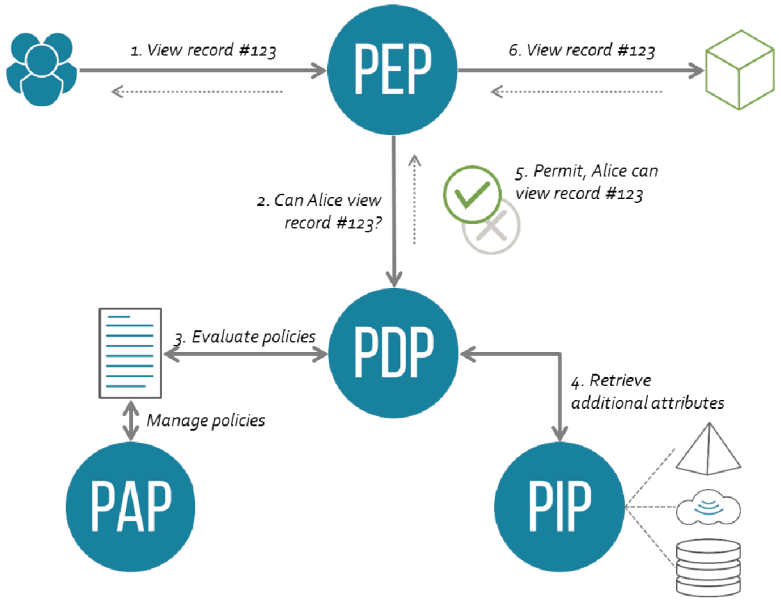
Das linhas 8 a 27 tem-se a declaração do elemento Target. Dentro do target são especificadas os sujeitos e a ação pretendida. Cada categoria de sujeito e ação ficam dentro de um Match, que recebe um AttributeValue. Para o exemplo, dentro do primeiro Match, a linha 14 informa que para sujeito *(urn:oasis:names:tc:xacml:1.0:subject:subject-id)* o valor esperado de AttributeValue é “alice”, linha 12.

O próximo Match, fará análise da entidade ação. Na linha 21 podemos ver a ação *(urn:oasis:names:tc:xacml:1.0:action:action-id)* esperada é “access”, especificada com AttributeValue na linha 19. O atributo XML obrigatório MustBePresent, determina se deve ser retornada uma string ou indeterminada caso o atributo especificado não seja encontrado na requisição. Para o sujeito e ação recebeu valores “false” conforme linhas 16 e 23.

Entre as linhas 28 e 64 está definido o elemento Rule, a regra que será analisada caso os valores dentro do target forem satisfeitas. Na linha 35, é possível notar que o recurso *(urn:oasis:names:tc:xacml:1.0:resource:resource-id)* que é tratado é o histórico, AttributeValue Histórico (History) na linha 33. A condição (Condition) para Alice acessar o histórico está entre as linhas 42 e 63. Lá é utilizado elementos Apply a fim de aplicar funções de comparação de tempo do XACML. Caso a hora atual *(urn:oasis:names:tc:xacml:1.0:environment:current-time)*, linha 47, do ambiente *(urn:oasis:names:tc:xacml:3.0:attribute-category:environment)*, linha 48, esteja entre os valores 08:00:00 e 18:00:00 horas resultará em Permit (permitir), definido no início da regra, linha 28. Caso contrário, o segundo Rule será avaliado na sequência, linha 65, com efeito de negar (Deny)

Como mencionado, Toktar, 2016 diz que a linguagem XACML é poderosa para padronizar trocas de mensagem request e response entre PEP e PDP. Abaixo, tem-se a figura 7 ilustrando a sequência de passos de uma requisição.

Figura 7 - ordem das requisições



Fonte: Silva, 2018

Quando um usuário deseja aplicar alguma ação sob um recurso, o PEP extrai os atributos e monta uma requisição em linguagem XACML Request até o PDP. O PDP é responsável por selecionar e aplicar as políticas definidas sobre os atributos da requisição, tomando a decisão de permitir ou negar a requisição em questão.

Nele é selecionada a política junto ao PAP de acordo com a requisição. Informações dos atributos das entidades envolvidas, como sujeito, ambiente e recurso, são fornecidos pelo PIP. O PDP tendo tomado uma decisão, envia uma resposta XACML Response ao PEP. Por fim, o PEP aplica a decisão tomada pelo PDP, permitindo ou negando o acesso solicitado pelo sujeito (SILVA, 2018).

A figura 8, é um exemplo de XACML Request. Na linha 2 os atributos do tipo booleano obrigatórios CombinedDecision e ReturnPolicyIdList. CombinedDecision é utilizado para informar ao PDP se ele deve combinar múltiplas decisões em uma única decisão. Já o atributo ReturnPolicyIdList especifica se a lista de políticas utilizadas para tomar a decisão deve ser retornada pelo PDP (SILVA, 2018). O restante da requisição é passado os valores dos atributos subject-id, action-id, resource-id e environment (current-time), sendo o horário atual de 12:36:25. O IncludeInResult, presente em todas as categorias do exemplo, é um atributo booleano que especifica se o atributo da entidade deve ser incluído na resposta da requisição.

Figura 8 - XACML Request

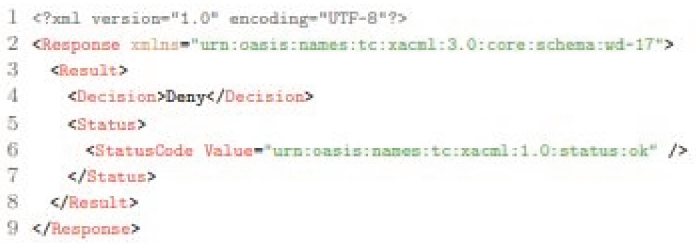
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Request CombinedDecision="false" ReturnPolicyIdList="false" xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
3    <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
4      <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id">
5        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">alice</AttributeValue>
6      </Attribute>
7    </Attributes>
8    <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
9      <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id">
10       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
11     </Attribute>
12   </Attributes>
13   <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
14     <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id">
15       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">history</AttributeValue>
16     </Attribute>
17   </Attributes>
18   <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
19     <Attribute IncludeInResult="false" AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time">
20       <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">12:36:25</AttributeValue>
21     </Attribute>
22   </Attributes>
23 </Request>
```

Fonte: Silva, 2018

A figura 9 apresenta um exemplo de XACML Response. Nesse exemplo o valor retornado pelo PDP é Deny. Na linha 2 é feita a declaração da resposta com o elemento Response e o atributo xmlns especificando a versão 3 do documento

XACML. O elemento Status é opcional e representa o status da requisição, indicando se ocorreu algum erro durante a avaliação da requisição e informações sobre o erro.

Figura 9 - XACML Response



```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Result>
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok" />
    </Status>
  </Result>
</Response>
```

Fonte: Silva, 2018

As linguagens de políticas estruturadas, como XACML, podem desempenhar um papel importante, principalmente se adotado um padrão universal de utilização. Enquanto muitas aplicações e plataformas têm seus próprios idiomas para controle de acesso, raramente são adequados para aplicar políticas de privacidade, e a falta de um padrão único torna a auditoria de conformidade um pesadelo. (ANDERSON, 2006, p. 1). A questão sobre adotar um padrão único na linguagem para políticas de privacidade é reforçada no artigo de Anderson (2006, p. 1), onde a linguagem usada para políticas de privacidade deve ser independente de plataforma e ser a mesma ou integrado à linguagem usada para políticas de controle de acesso, porque os dois tipos de políticas geralmente controlam o acesso aos mesmos recursos e não devem estar em conflito.

2.4 PRIVACIDADE BY DESIGN

A privacidade by design (PbD) é um conceito que introduz já nas etapas iniciais do desenvolvimento de software os princípios de privacidade. Segundo Pattakou (2018), existe um entendimento comum de que a privacidade requer atenção desde os estágios iniciais do sistema. Ainda segundo Pattakou (2018), muitos engenheiros de software já entenderam a necessidade e introduziram algumas metodologias nas etapas iniciais para modelar os requisitos de privacidade.

Contudo, pesquisas mostram que a cultura do PbD ainda é algo distante para alguns desenvolvedores. No artigo de Hadar (2018) foram realizadas entrevistas com 27 desenvolvedores de diferentes domínios a fim de entender a percepção deles quanto à privacidade das informações. Os resultados mostraram que, exceto no contexto de domínios específicos, os desenvolvedores estão ativamente desencorajado de fazer da privacidade uma prioridade. Muitos não priorizam pois não têm conhecimento suficiente e compreensão do conceito de privacidade ou desconhecem as tecnologias de preservação da privacidade (HADAR, 2018).

As metodologias associadas ao PbD são, na maioria, processos de desenvolvimento com foco nos dados dentro do software em construção. A metodologia LINDDUN, por exemplo, busca ter o desenho de um diagrama do fluxo de dados analisando quais atores terão acesso e a finalidade envolvida.

A metodologia SQUARE for privacy é orientada a riscos e auxilia na categorização e priorização dos requisitos de segurança em várias através da base de dados da ferramenta PRET.

Por fim, pode-se citar a metodologia STRAP que consiste em um processo iterativo de quatro etapas: análise, refinamento, avaliação e iteração. O objetivo é buscar uma análise estruturada das vulnerabilidades para auxiliar na modelagem dos requisitos de privacidade.

2.5 FERRAMENTA SECURITY POLICY TOOL (SPT)

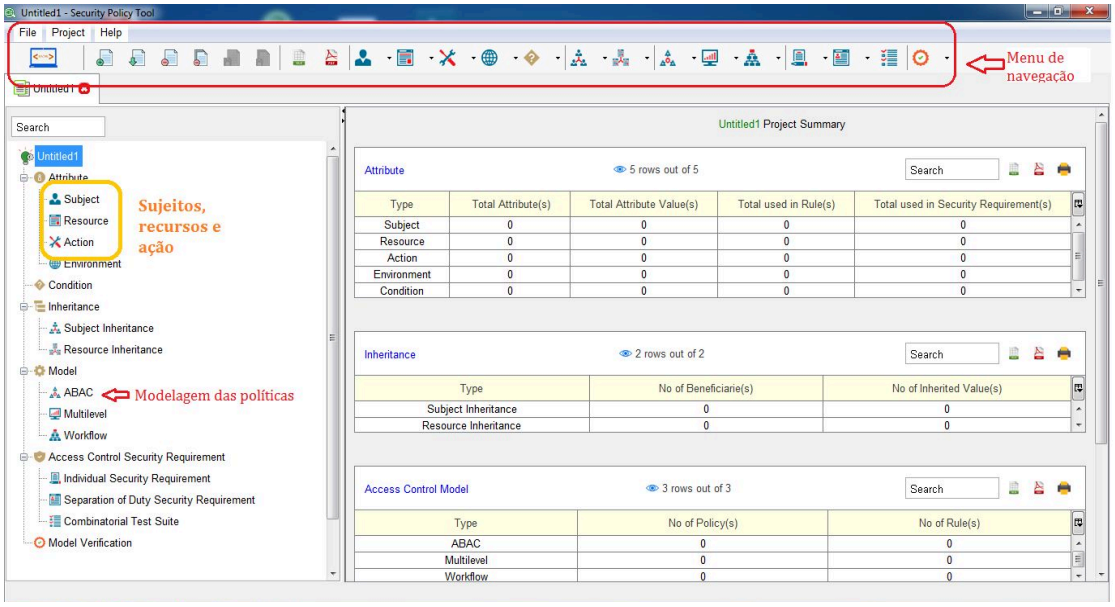
A ferramenta SPT oferece automatização e boa abrangência nos testes das políticas. Isso permite ter objetivos mais específicos no projeto, como identificar e corrigir possíveis erros nas políticas propostas e propor alterações se necessário. O site da ferramenta é bem completo. Dispõe de manual de usuário explicando todas as funcionalidades do software e exemplos de modelagem com documentação em PDF e vídeos explicativos com casos de testes, políticas no modelo ABAC e análise de resultados.

Para ter acesso ao software foi necessário criar um breve cadastro informando dados da instituição e título do trabalho de pesquisa no site oficial da ferramenta. Após isso o download é liberado. Na central de downloads do site

encontra-se instaladores para Windows, Linux e MacOS. A versão mais completa custa 180 dólares.

A ferramenta está disposta com um menu na esquerda onde é possível criar os sujeitos, recursos e regras, menu na parte de cima onde podemos acessar o editor XACML, abrir e fechar projetos e atalhos como segunda opção para criar sujeitos, recursos e regras. Na parte central é mostrado o detalhamento de cada recurso e regras. Temos imagem abaixo com visão geral da ferramenta:

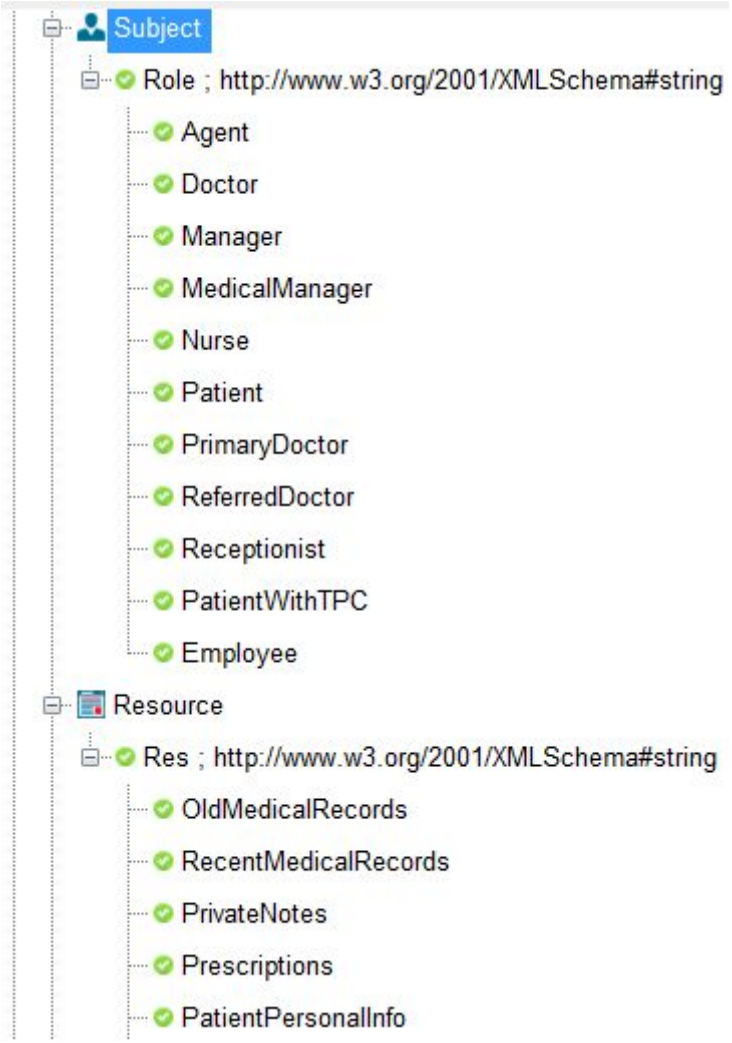
Figura 10 - tela inicial SPT



Fonte: Elaborada pelo autor

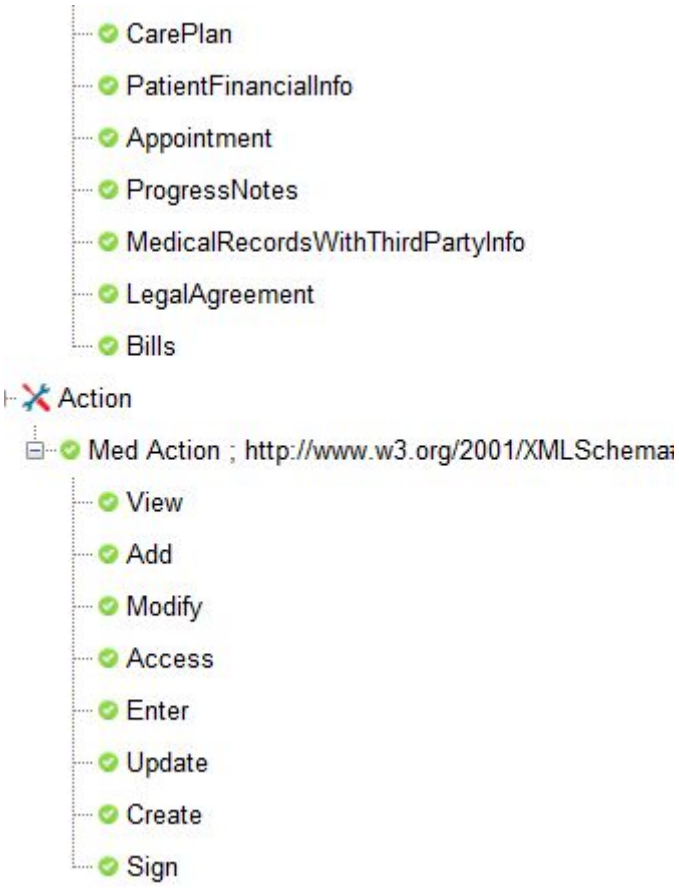
Na figura 11 e 12 abaixo temos os espaços de criação e configuração dos sujeitos, recursos e ações. Sujeitos, de acordo com as figuras abaixo, podem ser doutores, pacientes, recepcionistas entre outros possíveis. O recurso seria o objeto a ser acessado, como registros médicos antigos ou prescrições. Por fim, as ações possíveis de ver, adicionar, modificar, acessar, entrar, atualizar, criar e seguir.

Figura 11 - menu parte esquerda



Fonte: Elaborada pelo autor

Figura 12 - continuação menu esquerda



Fonte: Elaborada pelo autor

Tomando como exemplo, selecionamos o sujeito doutor. Abaixo na figura 13, podemos identificar as ações permitidas para cada recurso:

Rule(s) engaged with selected attribute (Role = Doctor): 7 rows out of 7 Search

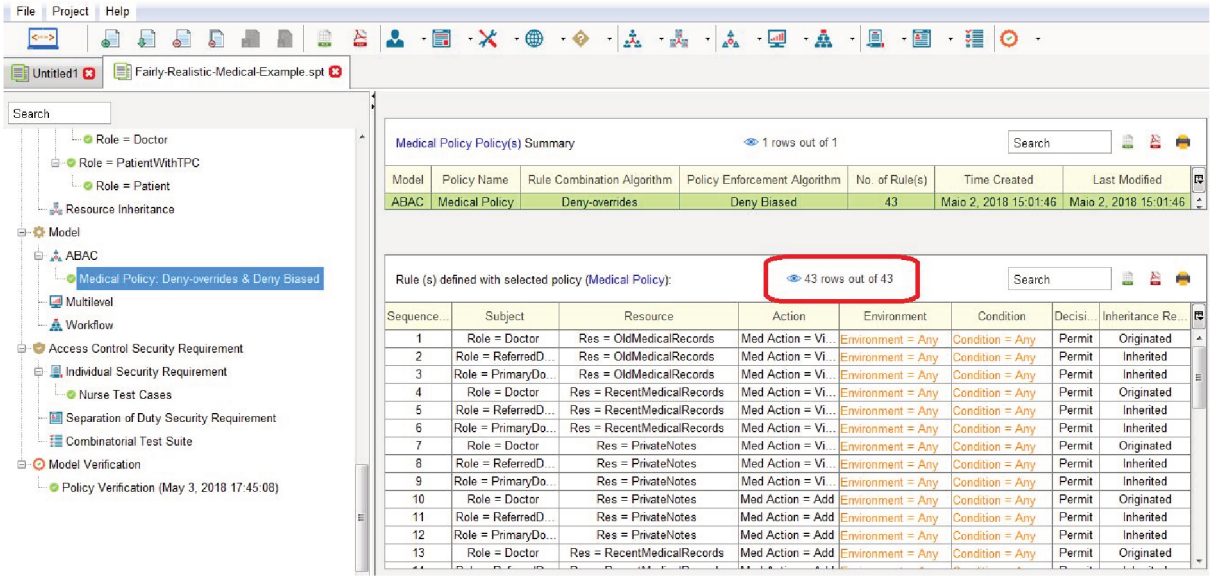
Model	Policy Name	Rule Combinatio...	Policy Enforcem...	Subject	Resource	Action	Environ...	...	Decision	In...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = OldMedicalRecords	Med Action = View	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = RecentMedicalRecords	Med Action = View	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = PrivateNotes	Med Action = View	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = PrivateNotes	Med Action = Add	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = RecentMedicalRecords	Med Action = Add	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = Prescriptions	Med Action = View	Environmen	Co	Permit	Ori...
ABAC	Medical Policy	Deny-overrides	Deny Biased	Role = Doctor	Res = Prescriptions	Med Action = Mod...	Environmen	Co	Permit	Ori...

Fonte: Elaborada pelo autor

Pode-se perceber que no exemplo é permitido ao doutor ver e modificar prescrições (*Prescriptions*), ver e adicionar notas privadas (*PrivateNotes*) e registros médicos recentes (*RecentMedicalRecords)* e somente ver registros médicos antigos (*OldMedicalRecords*).

Na figura 14 abaixo, temos o modelo ABAC e suas políticas associadas. Assim, no exemplo temos a política médica (Medical Policy) com 43 regras. Portanto, estão definidos os algoritmos utilizados e as condições dos valores e ambientes para permitir acesso a um recurso. Percebe-se também a informação se uma regra foi originada ou herdada que uma regra pai.

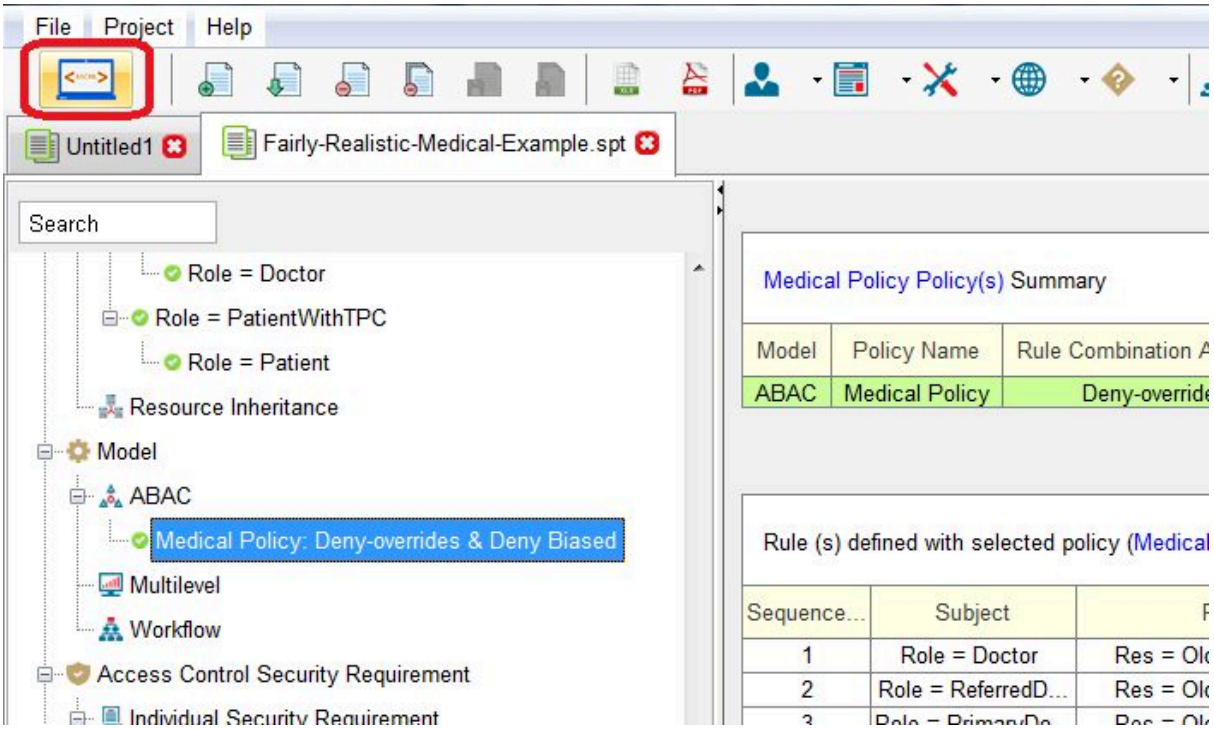
Figura 14 - regras modelo ABAC



Fonte: Elaborada pelo autor

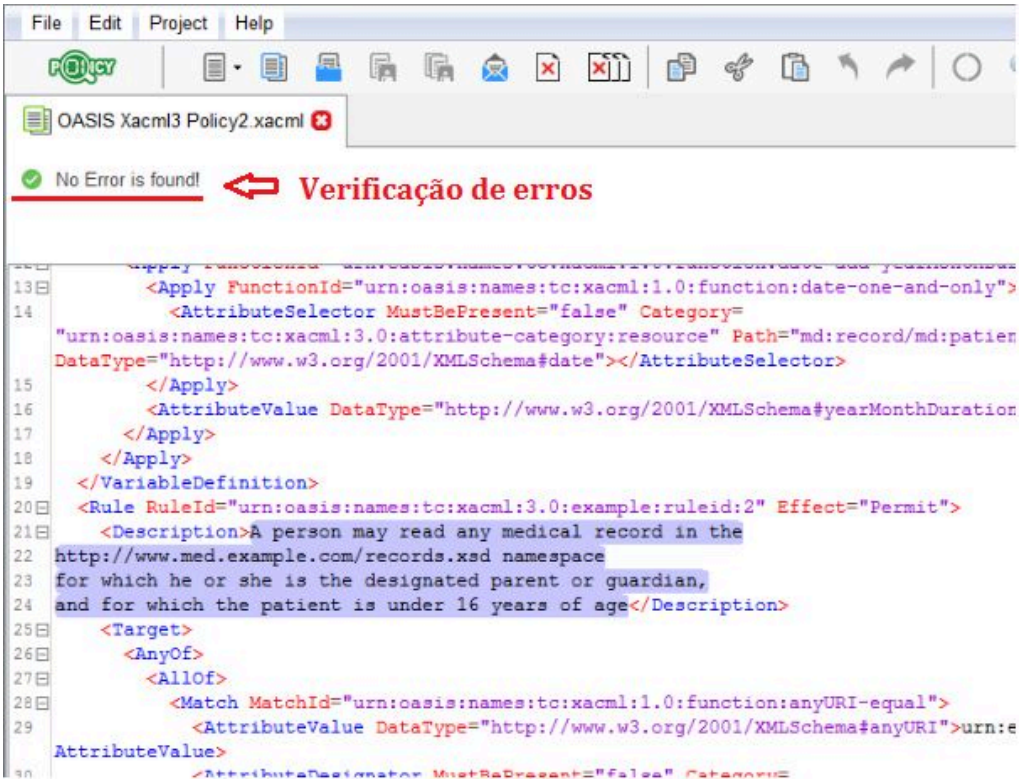
Um fator interessante é que mesmo tendo a disposição a interface intuitiva para a criação das regras ainda assim é possível trabalhar diretamente no código XACML através do botão destacado na figura 15 abaixo. A figura 16 contém um trecho de código em XACML da política modelada dentro da ferramenta. Destaca-se a funcionalidade de verificação de erros. Erros sintáticos no código são alertados ao usuário.

Figura 15 - editor XACML



Fonte: Elaborada pelo autor

Figura 16 - descrição regra XACML



Fonte: Elaborada pelo autor

2.6 FERRAMENTA SECTET-PL

O protótipo Sectet-PL utiliza um workflow baseado em UML (Unified Modeling Language) para gerar as políticas de controle de acesso em linguagem XACML. O administrador de rede faz a modelagem das permissões e a ferramenta se encarrega de traduzir e interpretar os dados para gerar o código para as políticas. Este gerador é destinado para a segurança de modelos de WorkFlow (CERETTA, 2007). O Sectet-PL foi projetado com o objetivo de gerenciar políticas de unidades hospitalares e não possui tratamento de conflitos.

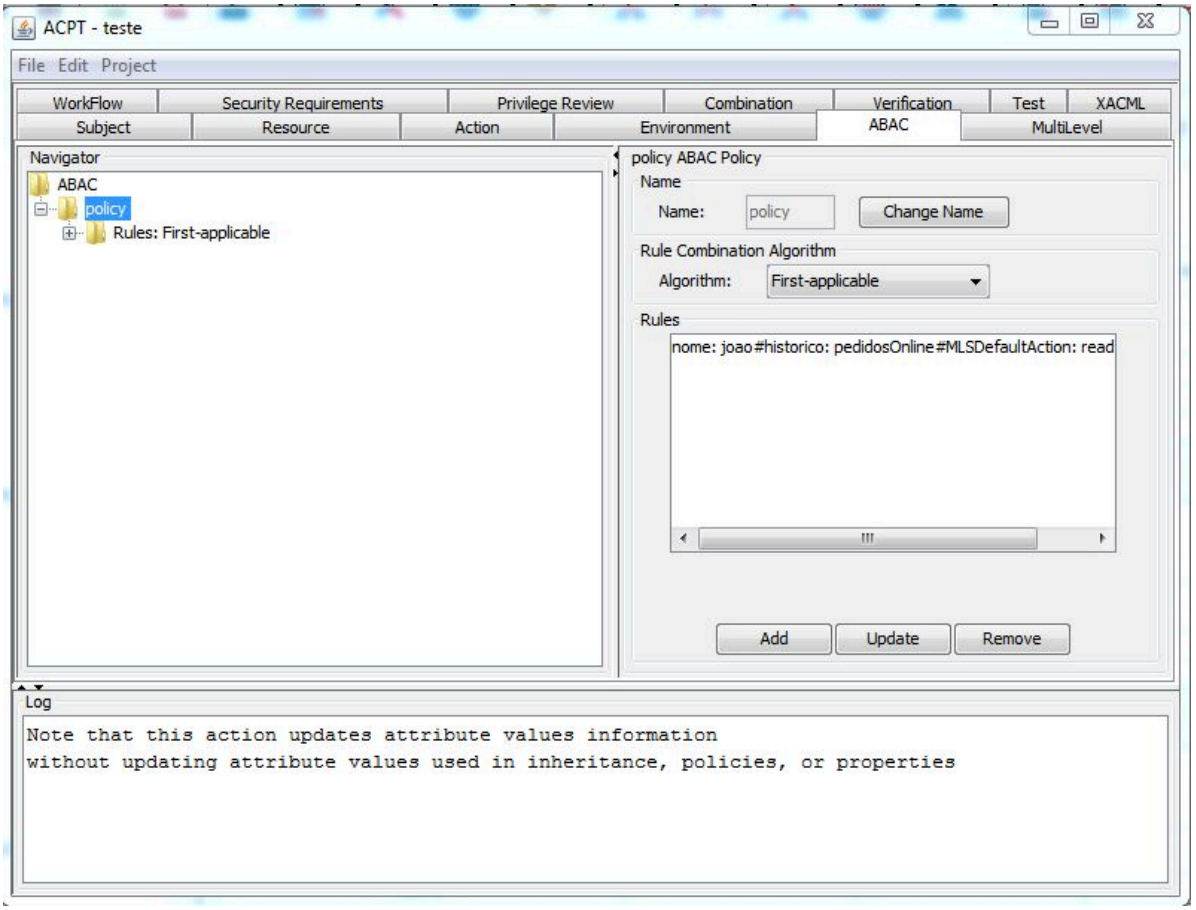
2.7 FERRAMENTA ACPT

O ACPT (Access Control Policy Testing) foi desenvolvido por 4 pesquisadores em 2010, um deles membro da divisão de segurança computacional do NIST. Para se ter acesso a ferramenta não é algo tão fácil. É necessário fazer um breve cadastro no site do NIST (National Institute of Standards and Technology) e posteriormente solicitar à divisão de segurança computacional do NIST/ITL, via e-mail, a senha para descompactar o arquivo de download. Pode-se considerar algo bom pelo fato do acesso a ferramenta ser controlado, remete mais segurança e confiabilidade ao pesquisador usuário da ferramenta.

O ACPT, desenvolvida em Java, é composta por três funcionalidades principais. A primeira ajuda a especificar e combinar políticas com base em modelos de políticas existentes. A segunda, o ACPT analisa e converte uma política em formato executável, como XACML. Em terceiro lugar, para assegurar a correção das políticas, a ACPT conduz a verificação estática e dinâmica de uma política executando um conjunto de testes (HWANG, 2010) através da integração com outro software NuSMV. A verificação estática consiste em analisar com o auxílio de uma máquina de estados todas as possibilidades de resultados de uma política. Em caso de falha em algum teste a ferramenta não exibe sugestões de correção, cabe ao autor identificar e corrigir (HWANG, 2010).

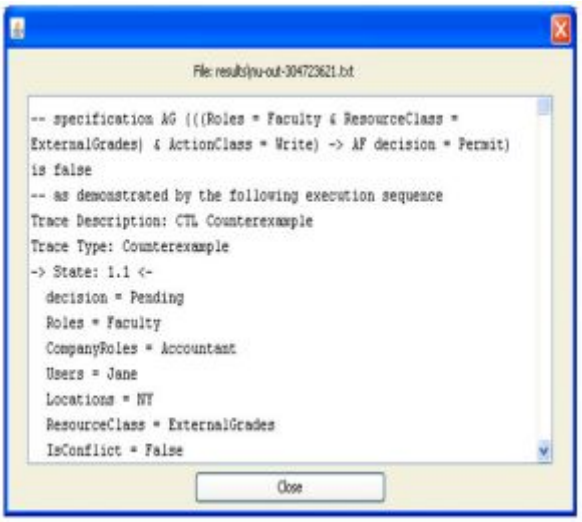
O ACPT fornece interface gráfica para ajudar os autores de políticas a especificar políticas e suas propriedades de maneira interativa e eficaz. Além disso, o ACPT também suporta recursos adicionais onde os autores de políticas podem editar, adicionar ou excluir políticas e seus atributos de forma interativa. A figura 17 mostra uma tela da ferramenta com suas funcionalidades:

Figura 17 - Ferramenta ACPT



Fonte: Elaborada pelo autor

Figura 18 - testes ACPT



Fonte: HWANG, 2010

2.8 REGULAMENTAÇÕES

Pelo mundo todo existem leis e normas que asseguram a privacidade e uso dos dados dos usuários. O NIST por exemplo possui alguns guias com boas práticas para verificação e teste de políticas de controle de acesso, a NIST SP 800-192. Outro importante é a NIST IR 7316 para verificação de sistemas de controle de acesso.

Aqui no Brasil a Lei Geral de Proteção de Dados (LGPD), sancionada em agosto de 2018, entrará em vigor em agosto de 2020. Servirá para regular o uso, coleta, armazenamento e divulgação dos dados pessoais. A lei, exigirá das empresas e órgãos do governo maior segurança observando a finalidade, adequação, prevenção, transparência e responsabilização. A nova legislação exigirá também consentimento para o uso prevendo penalidades em caso de descumprimento. O primeiro avanço do país nesse sentido foi com o Marco Civil da Internet, Lei n° 12.965 de 2014.

A LGPD teve como origem na RGPD (Regulamento Geral sobre a Proteção de Dados) da União Europeia, que possui um propósito semelhante aplicado na europa. Pode-se citar também a CCPA (Califórnia Consumer Privacy Act), de 28 de junho de 2018 como uma importante legislação do estado da Califórnia.

2.9 TRABALHOS RELACIONADOS

Ao pesquisar sobre trabalhos relacionados foram identificados uma tese de doutorado, Werner (2017) e um artigo de Mont (2006).

A tese de Werner (2017) propõe um modelo de privacidade para o gerenciamento de identidade em ambientes de nuvem. Além de todos os conceitos trazidos pela tese e as contribuições para o estado da arte, foi implementado um protótipo a fim de visualizar na práticas os conceitos de privacidade propostos em ambiente de nuvem. Foram criados alguns casos de uso. Os estudos de casos nas áreas médica e em comércio eletrônico demonstraram que a presença de mecanismos, negociação e políticas é essencial na administração de dados sensíveis (WERNER, 2017).

O capítulo mais importante para o contexto deste trabalho foi o 6, Políticas no Gerenciamento de Identidade. Foram apresentadas definições para que as preferências de privacidade do usuários fossem definidas. Essas definições é que serviram como base de estudo para implementação na ferramenta escolhida.

O artigo de apresenta uma solução para automatizar a aplicação da privacidade nas empresas de maneira sistêmica, controlando o acesso aos dados pessoais sensíveis a privacidade. O protótipo apresentado propõe um modelo que seria capaz de interceptar uma consulta no banco de dados, transformá-la numa query compatível com a original levando em conta as restrições de privacidade declaradas para então ser executada no banco de dados. No artigo, foi feita uma integração desta solução com o HP Select Identity, uma solução de ponta para gerenciar identidades digitais dentro e entre grandes empresas (MONT; THAYNE, 2006). O resultado foi satisfatório e concluído que está apto a ser explorada comercialmente.

3 - APRESENTAÇÃO DAS DEFINIÇÕES

Na tese de Werner (2017, p. 129) temos a formalização das políticas apresentadas na tese distribuídas em 6 definições.

3.1 - DEFINIÇÃO 1

A primeira, é retratado o conceito de serviço. É definido por um possível conjunto de tipos de entrada, uma função que produz uma saída para cada entrada, e um conjunto de possíveis tipos de saída (WERNER, 2017). Em suma, todo serviço requisitado requer um conjunto de dados de entrada, na qual, serão processados para produzir um resultado. Um provedor de serviços exemplificado nessa mesma definição é a de transporte. O usuário fornece dados como nome e endereço, o provedor processa essa informação e retorna como saída um código de rastreio.

3.2 - DEFINIÇÃO 2

Para o nosso contexto, deve-se atentar sobre a privacidade, utilização e disseminação desses dados de entrada. A definição 2 aborda de forma mais objetiva esse contexto. Na tupla de itens contemplados estão contidos as seguintes informações:

- O tipo de dados para o qual a regra de política é definida (dT)
- conjunto de propósitos a se usar o dT (pR)
- tempo de conservação dos dados dT em horas (tM)
- o contexto especificado para determinado tipo de dados dT (cN)
- interesse na notificação sobre o uso dos dados (nT)
- interesse na cifragem dos dados (cP)
- interesse em obrigações a cumprir (oB)

A equação resultante da definição 2:

$$ProvedorServicos.RegrPrivacidade = \{dT; pR; tM; cN; nT; cP; oB\}$$

Segundo Villareal (et. al, 2017) citado por Werner, 2017, os tipos de dados dT podem assumir valores de 5 categorias de informações de identidade pessoal (PII) possíveis. São elas:

- Identificação Pessoal (IP) – abrange qualquer tipo de informação que represente o dono do PII, por exemplo, nome, identificadores nacionais, nomes dos pais, endereço residencial, foto e número do cartão de crédito;
- Preferências e Características Pessoais (PCP) – são considerados como os atributos físicos dos donos dos PIIs e as suas opções pessoais, por exemplo, peso, crenças religiosas e orientação sexual;
- Localização (LO) – se refere a qualquer informação sobre onde o usuário está ou foi e suas trajetórias com qualquer grau de precisão e obtidos por quaisquer meios, por exemplo, GPS, redes sem fio ou sistemas de telecomunicações;
- Atividades e Hábitos (AH) – são quaisquer atividades realizadas pelo usuário e hábitos inferidos, por exemplo, páginas visitadas, compras e perfil comportamental;
- Relacionamentos (RS) – são as pessoas com quem o outorgante das PII está em um momento específico, ou interage através de meios, como por exemplo, através de redes sociais, e-mails e mensageiros instantâneos.

Os propósitos também são categorizados:

- Melhoria de Serviço (MS) – refere-se ao uso dos dados para a implementação de melhorias nos serviços oferecidos, como personalização de funcionalidades, maior usabilidade e aumento da segurança;
- Científico (CI) – usuários que desejam disponibilizar dados apenas para pesquisas científicas;
- Comercial (CO) – o compartilhamento de dados com propósito de fins comerciais, por exemplo, compras pela Internet;
- Governamental (GO) – o compartilhamento de dados com propósito de fins governamentais, por exemplo, previdência social.

Para os contextos possíveis, tem-se as seguintes categorias:

- Financeiro (FI) – ambiente de banco;
- Compras online (EC) – ambiente de comércio eletrônico;
- Saúde (SA) – ambiente médico;

- Militar (MI) – instalações militares

Para os atributos de notificação sobre o uso dos dados (nT) e cifragem dos dados (cP) os valores possíveis são “1” para sim ou “0” para não.

3.3 - DEFINIÇÃO 3

A definição 3 é retratado a preferência de privacidade do usuário para os dados de entrada. Um tipo de dado dT atrelado a um nível de sensibilidade de privacidade, bem como o propósito destinado. O nível pode ter valores de 1 a 3. Sendo 1 nível baixo, 2 médio e 3 nível alto. O detalhamento dos níveis é apresentado na seção 5.2.2 do artigo de WERNER, 2017 baseado em perfis de usuários e nos atributos que identificam diretamente uma pessoa. A equação é definida:

ProvedorServicos.PrefPrivacidade = [dT, Nivel, pR]

3.4 - DEFINIÇÃO 4

A definição 4 é propriamente a política de privacidade do usuário. São passadas 3 elementos:

- os dados, os atributos do usuário
- a regra de privacidade ao disseminar os dados ao serviço, seria a definição 2.
- preferência de privacidade sobre os dados de entrada, já vistas na definição 3.

  A equação é definida:

  ProvedorServicos.PoliticaPrivacidadeUsuario = Dados, RegrPrivacidade, PrefPrivacidade

As definições 5 e 6 tratam mais especificamente a solicitação e controle de acesso. Os atributos de suas equações já foram apresentadas pelas definições

anteriores e seria redundante sua representação no contexto e objetivo do presente trabalho.

4 - MODELAGEM DAS DEFINIÇÕES DE PRIVACIDADE NA FERRAMENTA SPT

A versão da ferramenta SPT utilizada no desenvolvimento do presente trabalho é gratuita e, assim, contém limitações. Algumas modelagens apresentadas neste capítulo foram adaptadas a fim de contornar tais limitações porém sem impactar no entendimento e resultado esperado. A maior limitação é ter no máximo 3 classes para sujeitos, recursos e ações. Algumas representações foram feitas como sub-classes.

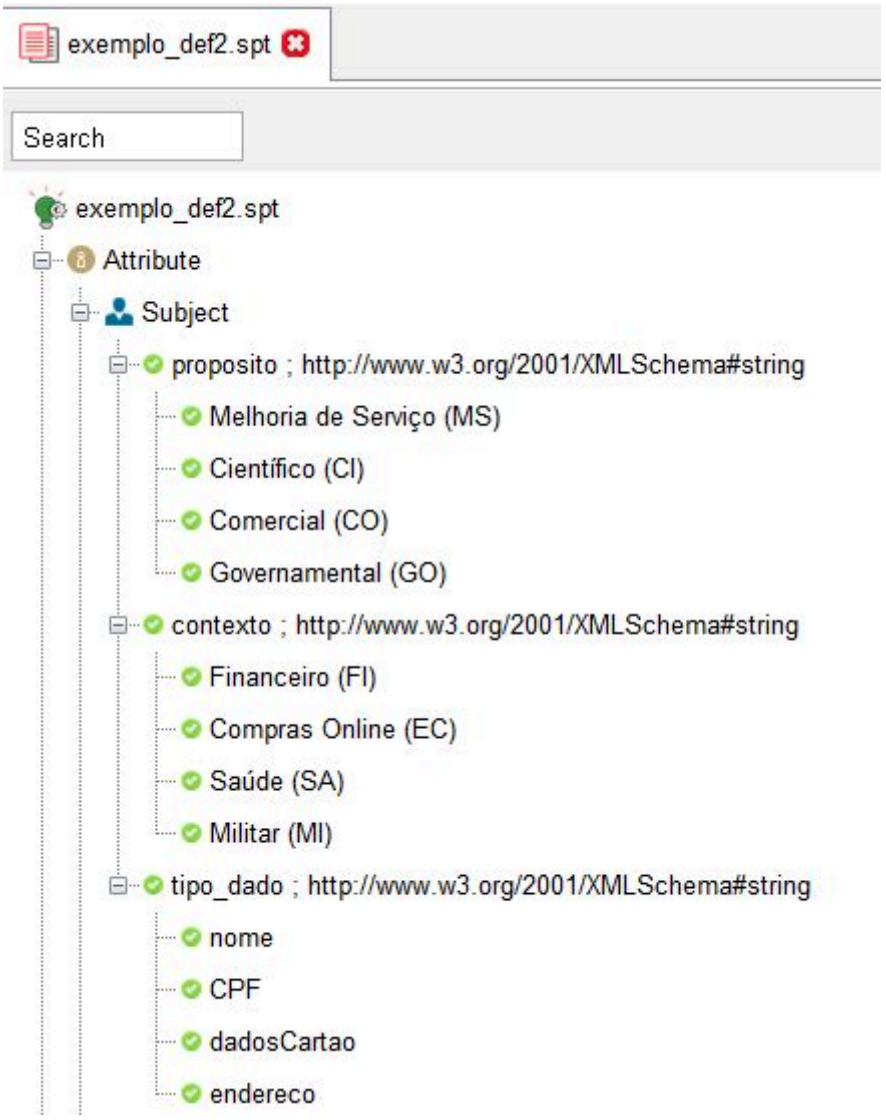
4.1 MODELAGEM DA DEFINIÇÃO 2

A primeira modelagem feita na ferramenta foi a definição 2. O intuito é observar na prática com exemplificações das definições e fazer uma verificação do modelo com testes. No artigo de Werner, 2017 é retratado nesta definição um exemplo de regra de privacidade que diz que os tipos de dados (nome e CPF) podem serem usados apenas para fins de pagamento. O período de tempo para manter um valor de tipo nome e CPF é de 48 horas, apenas podem ser utilizados no contexto de compras online se o usuário deseja ser notificado sobre o uso dos dados, os dados devem ser transmitidos de forma cifrada e o sistema deve conferir o certificado de compra segura. A formalização do exemplo é representado:

SP.RegrP riv = [nome, cpf; pagamento; 48; compras; 1; 1; selo]

No entendimento do problema, foi mapeado os sujeitos da seguinte maneira, conforme figuras 19, 20 e 21:

Figura 19 - sujeitos definição 2

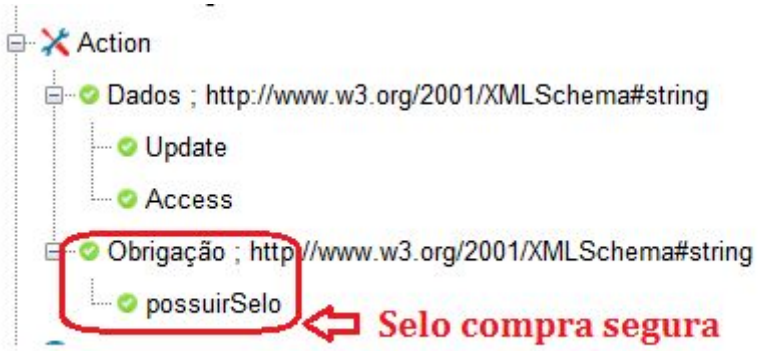


Fonte: Elaborada pelo autor

Figura 20 -recursos definição 2

Fonte: Elaborada pelo autor

Figura 21 -Ação definição 2



Fonte: Elaborada pelo autor

O atributo obrigações a cumprir (oB) foi modelado dentro de ações. A obrigação no exemplo refere-se ao provedor de serviços possuir um selo de compra segura.

Dentro de Condition (Condições), figura 22 mostra que o atributo tempo de conservação dos dados, foi mapeado no SPT como um valor booleano. Sugere-se que se a requisição ainda está dentro do prazo definido pelo usuário, recebe valor true, senão, valor false. Mesma regra se aplica para notificação e criptografia. True caso a variável receba valor “1” e False caso receba valor “0”.

Figura 22 - condição definição 2

Fonte: Elaborada pelo autor

No modelo utilizamos o ABAC combinando os algoritmos Deny-overrides & Deny Biased descrevendo o comportamento da regra de privacidade. Lá são definidas as condições para os recursos serem acessados.

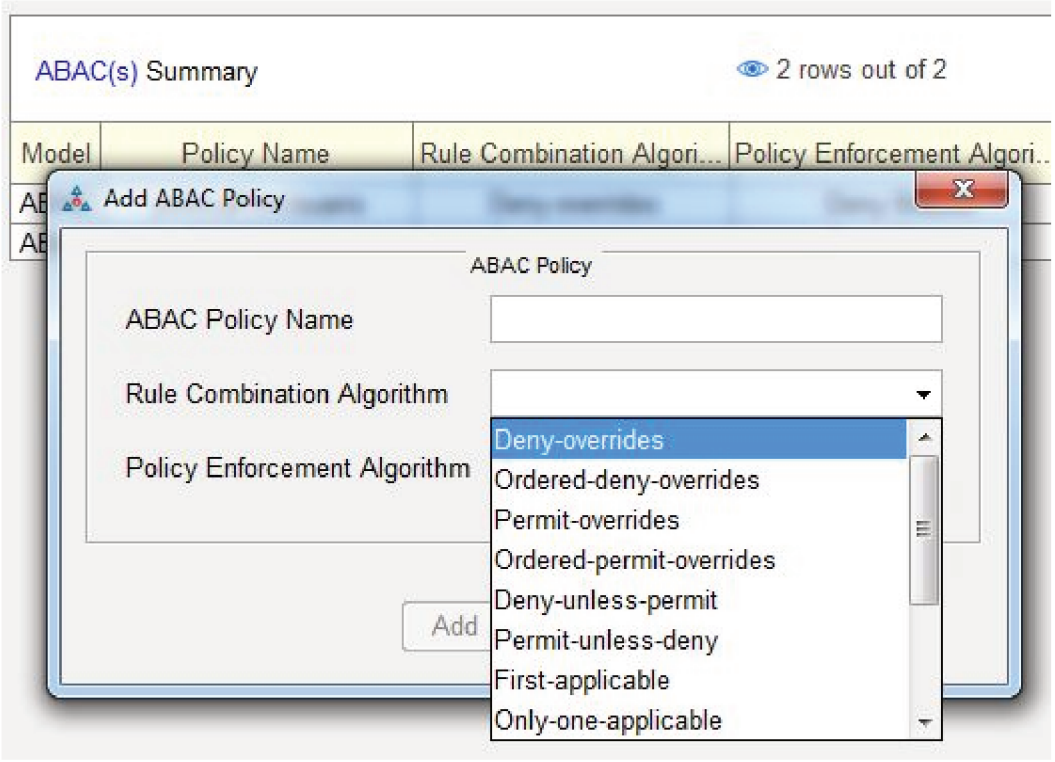
A figura 23 mostra que o SPT oferece outros algoritmos opcionais, além dos quatro tradicionais oriundos do XCML, que auxiliam no caso de alguns conflitos (INFOBEYOUND, 2017). Cabe aqui mencioná-los:

- Deny Unless Permit: esse algoritmo permite apenas duas decisões: Permitir ou Negar. Isto é destina-se aos casos em que uma decisão de permissão deve ter prioridade sobre uma decisão de negação, e um Indeterminado ou Não Aplicável nunca deve ser o resultado.
- Permit Unless Deny: semelhante ao algoritmo anterior porém com prioridade para negação sobre um caso de permissão. Indeterminado ou Não Aplicável também nunca serão válidos.
- Weak-Consensus: o consenso fraco exige que as políticas não entrem em conflito com as outras. Ele nega uma solicitação de acesso se algumas políticas negarem a solicitação e nenhuma política permitir. Semelhantemente, ele permite uma solicitação de acesso se algumas políticas permitirem e nenhuma política negar. Porém, produz conflito se algumas políticas permitirem e outras negarem.
- Strong-Consensus: esse algoritmo exige que todas as políticas concordem com uma decisão. Ela nega uma solicitação de acesso se todas as políticas negarem a solicitação. Permite uma solicitação de acesso se todas as

políticas permitem a solicitação. Caso contrário, o conflito é gerado.

- Weak-Majority: quando políticas diferentes tomam decisões conflitantes sobre uma solicitação, a solicitação será permitida se o número de políticas permitir for maior que o número de políticas negar.
- Strong-Majority: a maioria forte permite uma solicitação se mais da metade de todos políticas também permitem.
- Super-Majority-Permit: a permissão de super-maioria permite uma solicitação de acesso se mais de ⅔ de todas as políticas também permitam.

Figura 23 -algoritmo definição 2

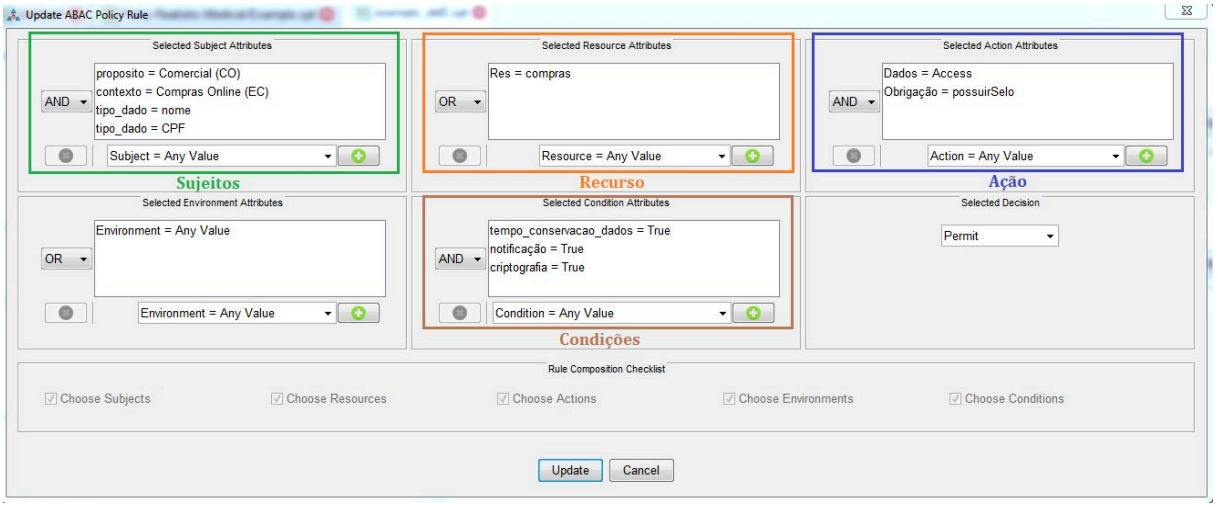


Fonte: Elaborada pelo autor

A categoria Policy Enforcement Algorithm tratam os casos com resultado Não Aplicável. Ou seja, quando a solicitação recebida não corresponde a nenhuma regra da política de controle de acesso, ela será permitida se for permit-biased, se for deny-biased, será negada.

A figura 24 retrata a montagem da política ABAC da definição 2. Os sujeitos escolhidos foram propósito comercial, contexto de compra online (EC), tipo_dado, nome e CPF. Recurso é compras. Ação é acessar o dado com a obrigatoriedade do selo de compra segura. Em relação as condições, o tempo_conservação_dados foi considerado True por estar dentro do valor de 48 horas, notificação e criptografia True. Após a implementação do exemplo pode-se iniciar os casos de testes.

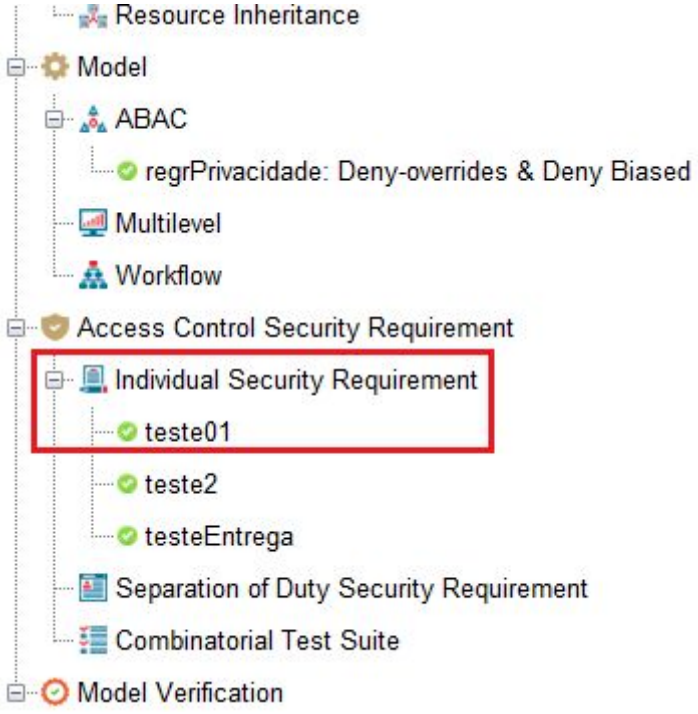
Figura 24 - descrição política definição 2



Fonte: Elaborada pelo autor

Em Individual Security Requirement, evidenciado na figura 25, é possível criar diversos casos de testes considerando diferentes requisições de acesso e para cada um deles especificar os valores para sujeitos, recursos, ações e as condições.

Figura 25 - Casos de teste definição 2

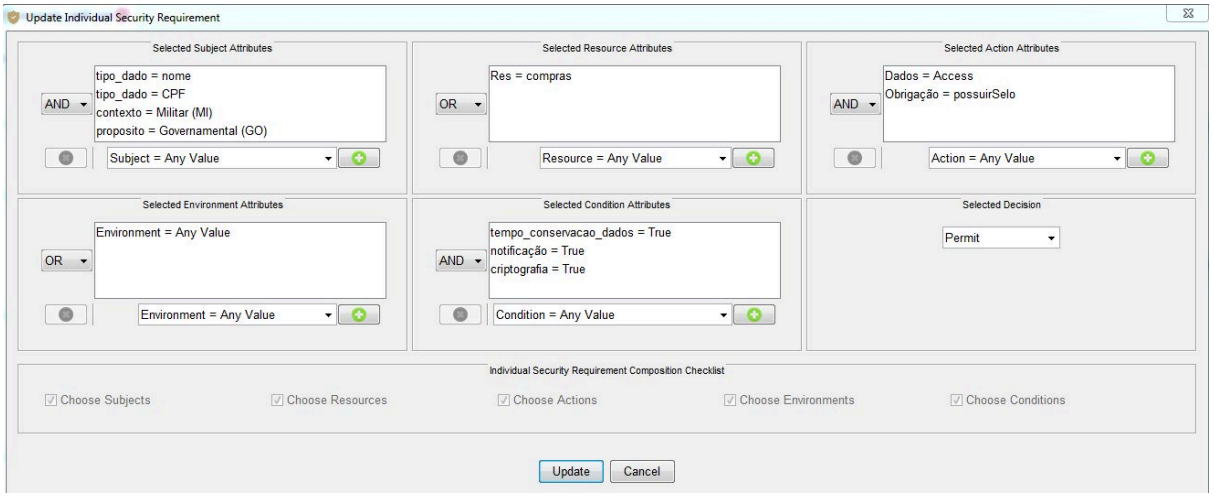


Fonte: Elaborada pelo autor

Esses casos serão ser selecionados mais adiante em Model Verification onde serão confrontados com as políticas. O teste01, figura 26, foi considerado que um

provedor de serviço gostaria de acessar os dados do usuário nome e CPF para um contexto militar e um propósito governamental. Tal situação é divergente com a política implementada no modelo ABAC e espera-se que o teste falhe. Abaixo tem-se os valores de atributos para o teste01:

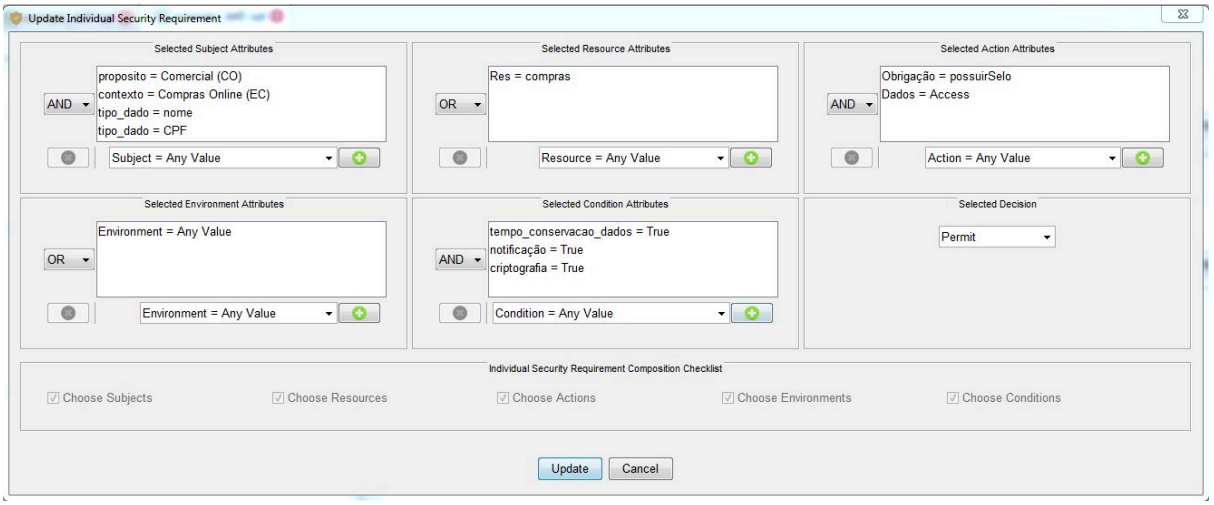
Figura 26 - descrição caso de teste 01



Fonte: Elaborada pelo autor

Da mesma forma, criou-se o teste2 com os valores de atributos corretos segundo a política ABAC da definição. A imagem 27 abaixo mostra como foi configurado:

Figura 27 - descrição caso de teste 02

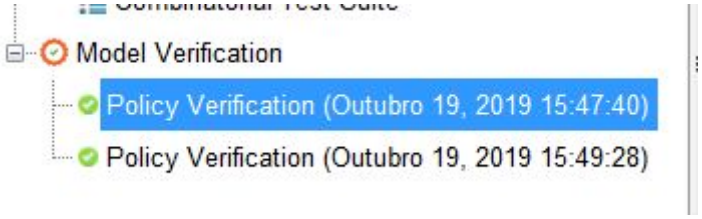


Fonte: Elaborada pelo autor

A funcionalidade Model Verification, destacada na figura 28, permite selecionar uma política, ou um conjunto delas, com um caso de teste a fim de

analisar os resultados. Nessa funcionalidade da ferramenta, são realizadas todas as combinações possíveis para TRUE e FALSE. Alguns casos podem receber valor Not Applicable quando não há regra para um determinado recurso.

Figura 28 - model verification definição 2



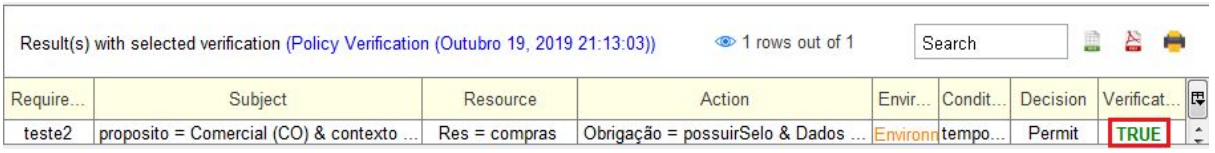
Fonte: Elaborada pelo autor

É comum nessa fase a percepção de conflitos entre regras. Caso ocorra, isso é possível resolver criando outras regras ou alterando os algoritmos de combinação do ABAC. Abaixo temos as figuras 29 e 30 com o resultado do teste01 e teste2 respectivamente. Como era esperado, o teste01 falhou nos testes e recebeu FALSE. Já o teste2 recebeu TRUE por estar conformidade com a política.

Figura 29 - resultado teste 01

Fonte: Elaborada pelo autor

Figura 30 - resultado teste 2

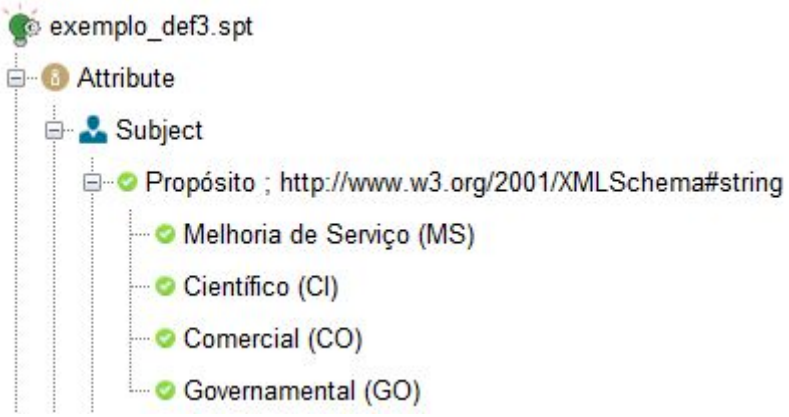


Fonte: Elaborada pelo autor

4.2 MODELAGEM DA DEFINIÇÃO 3

A modelagem da definição 3 teve algumas alterações em sua estrutura. O sujeito passou a ter propósito, o tipo_dado recebeu novos valores e foi acrescido os níveis de sensibilidade de privacidade como uma categoria de sujeito. Abaixo tem-se as figuras 31, 32 e 33 com valores para sujeito, recurso e ação:

Figura 31 - sujeito definição 3



Fonte: Elaborada pelo autor

Figura 32 - sujeito definição 3 continuação

Fonte: Elaborada pelo autor

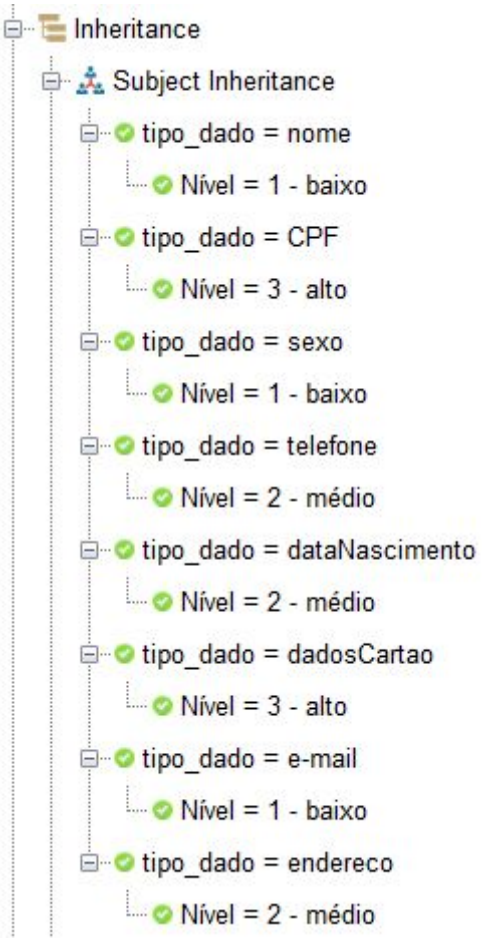
Figura 33 - recurso e ação definição 3

Fonte: Elaborada pelo autor

Dentro de Inheritance (herança), foi implementado uma pré-definição em relação níveis de privacidade para com cada tipo_dado do usuário (figura 34). O CPF, por exemplo, terá quando utilizado, nível 3 de privacidade. Em uma aplicação, os níveis poderiam ser mutáveis pelos usuários dependendo da profundidade da informação a ser acessada.

Na ferramenta SPT, a herança também pode ser construída graficamente conforme mostra a figura 35 do apêndice A. Os sujeitos beneficiados são ligados aos níveis que serão herdados.

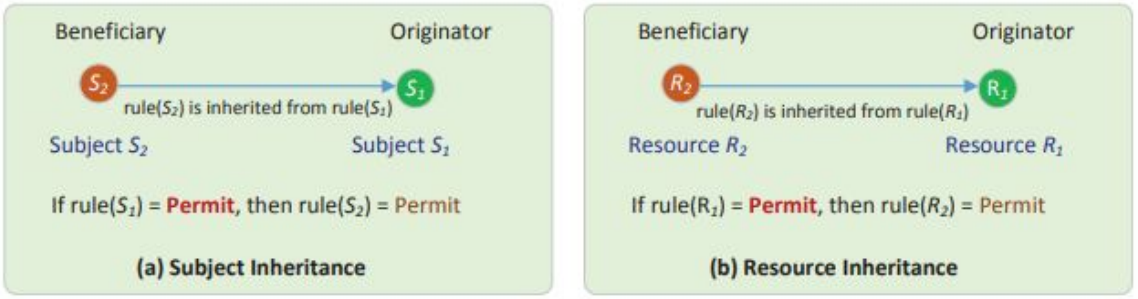
Figura 34 - herança dos níveis definição 3



Fonte: Elaborada pelo autor

A herança na ferramenta pode ser explicada pela figura 36 retirada do manual de usuário do SPT. Em organizações com muitos recursos e sujeitos torna-se mais fácil de compor as políticas em personagens com mesmo nível hierárquico ou com mesmos privilégios de acesso.

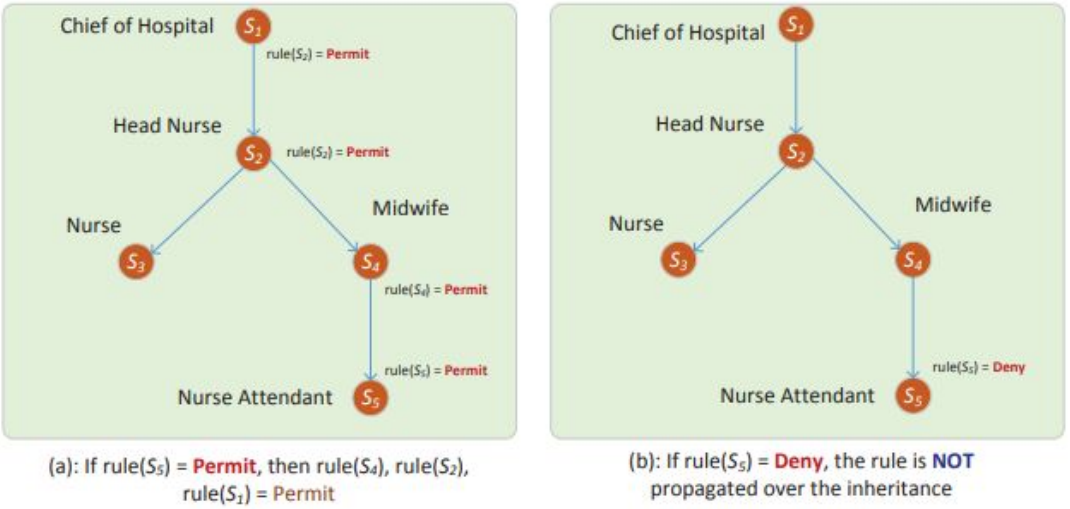
Figura 36 - detalhes herança SPT



Fonte: INFOBEYOUND, 2017

Este artifício propicia aumento de produtividade sem ter que criar regras redundantes bem como melhora a consistência das políticas. Com um número menor de regras, diminui-se a possibilidade de erro de regra ao editar várias regras, sobretudo redundantes (INFOBEYOUND, 2017). Na figura 36, o sujeito S2 herda os privilégios do sujeito S1. Ou seja, se uma regra aplicada ao sujeito S1 resultar em “permit”, para o sujeito S2, sob a mesma regra, também resultaria em “permit”. Contudo, o manual de usuário da ferramenta destaca que se um sujeito receber a decisão de negação (Deny), a propagação não ocorre para o nível acima. A herança não faria sentido se um nível superior herdasse uma negação de nível hierárquico inferior. A situação é exemplificada na figura 37 abaixo.

Figura 37 - negação em herança SPT



Fonte: Infobeyound, 2017

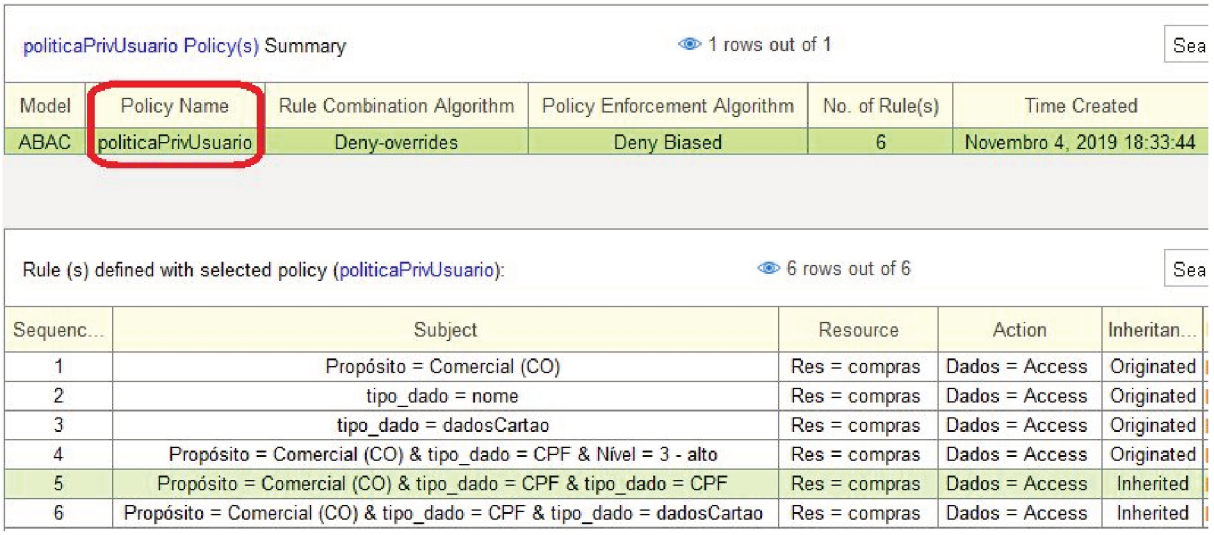
Isso posto, a definição 3 foi contida na ferramenta. O próximo passo, está descrito no subcapítulo seguinte com a modelagem da política de privacidade do usuário, a definição 4.

4.3 MODELAGEM DA DEFINIÇÃO 4

A definição 4, já explicado no capítulo anterior, é composta por 3 elementos: dados de entrada, regra de privacidade (definição 2) e preferência de privacidade do usuário (definição 3). Tendo as definições 2 e 3 modeladas, pode-se agora escrever

as políticas de uso dos atributos do usuário para um provedor de serviço. Na figura 38, temos a política 4 com suas regras:

Figura 38 - política definição 4



politicaPrivUsuario Policy(s) Summary — 1 rows out of 1

Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created
ABAC	politicaPrivUsuario	Deny-overrides	Deny Biased	6	Novembro 4, 2019 18:33:44

Rule (s) defined with selected policy (politicaPrivUsuario): — 6 rows out of 6

Sequenc...	Subject	Resource	Action	Inheritan...
1	Propósito = Comercial (CO)	Res = compras	Dados = Access	Originated
2	tipo_dado = nome	Res = compras	Dados = Access	Originated
3	tipo_dado = dadosCartao	Res = compras	Dados = Access	Originated
4	Propósito = Comercial (CO) & tipo_dado = CPF & Nível = 3 - alto	Res = compras	Dados = Access	Originated
5	Propósito = Comercial (CO) & tipo_dado = CPF & tipo_dado = CPF	Res = compras	Dados = Access	Inherited
6	Propósito = Comercial (CO) & tipo_dado = CPF & tipo_dado = dadosCartao	Res = compras	Dados = Access	Inherited

Fonte: Elaborada pelo autor

No topo da imagem temos o nome da política ABAC em destaque. Considera-se aqui que as regras de privacidade já foram configuradas pelo usuário para o serviço e que já estão incluídas na *politicaPrivUsuario*. O exemplo a seguir simboliza políticas e regras que o usuário destina o uso de seus dados bem como a preferência de privacidade para cada atributo. Pode-se observar que na linha de sequência número 1 o recurso compras só pode ser acessado em um propósito comercial. As linhas 2 e 3 diz que os tipos de dados permitidos são Nome e dados do cartão. A linha 4 é um pouco mais específica. O CPF, quando usado em propósito comercial, deve ser nível 3 de privacidade. Por esse motivo, as linhas 5 e 6 foram geradas por herança. Quando mencionado na regra anterior que o nível deve ser 3, automaticamente criou-se regras para outros tipo_dado de nível 3. No caso, CPF e dadosCartao. Em suma, é possível acessar o recurso compras apenas com nome ou dados do cartão. Porém quando usado CPF, só é possível com nível 3, seja por herança ou não. A figura 39 mostra que ao fornecer outro nível diferente do que 3 para propósito comercial, a requisição falha.

Figura 39 - resultado caso de teste nível errado

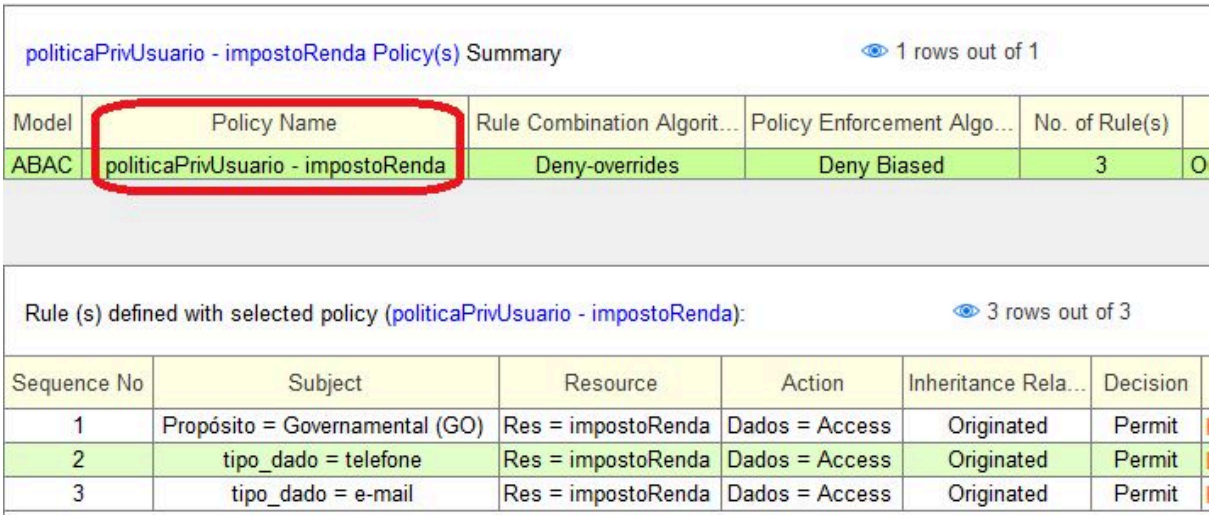
Result(s) with selected verification (Policy Verification (Novembro 4, 2019 19:31:01)) 1 rows out of 1

Requirement Sch...	Subject	Resource	Action	Verification Result
nivelErrado	Propósito = Comercial (CO) & tipo_dado = CPF & Nível = 2 - médio	Res = compras	Dados = Access	FALSE

Fonte: Elaborada pelo autor

Semelhantemente, segue um exemplo onde o usuário pode criar uma política de uso de seus dados para o serviço de imposto de renda. Para isso, é predefinido que para o propósito governamental, podem ser usados telefone e e-mail com seus níveis de sensibilidade herdados, 2 e 1 respectivamente. Abaixo a figura 40.

Figura 40 - política para imposto de renda



politicaPrivUsuario - impostoRenda Policy(s) Summary 1 rows out of 1

Model	Policy Name	Rule Combination Algorit...	Policy Enforcement Algo...	No. of Rule(s)	
ABAC	politicaPrivUsuario - impostoRenda	Deny-overrides	Deny Biased	3	O

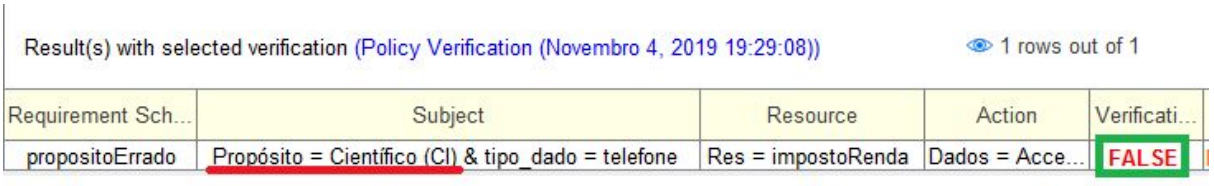
Rule (s) defined with selected policy (politicaPrivUsuario - impostoRenda): 3 rows out of 3

Sequence No	Subject	Resource	Action	Inheritance Rela...	Decision
1	Propósito = Governamental (GO)	Res = impostoRenda	Dados = Access	Originated	Permit
2	tipo_dado = telefone	Res = impostoRenda	Dados = Access	Originated	Permit
3	tipo_dado = e-mail	Res = impostoRenda	Dados = Access	Originated	Permit

Fonte: Elaborada pelo autor

A figura 41 retrata um caso de teste que é utilizado o telefone para propósito científico (CI). Em uma solicitação de acesso, teria resposta Deny (negação).

Figura 41 - resultado caso de teste propósito errado



Result(s) with selected verification (Policy Verification (Novembro 4, 2019 19:29:08)) 1 rows out of 1

Requirement Sch...	Subject	Resource	Action	Verificati...
propositoErrado	Propósito = Científico (CI) & tipo_dado = telefone	Res = impostoRenda	Dados = Acce...	FALSE

Fonte: Elaborada pelo autor

5 - ANÁLISE DOS RESULTADOS

Foi descrito no capítulo 3 as definições de privacidade que serviram como base para a pesquisa. Com isso, pode-se no capítulo 4 testá-las na ferramenta SPT demonstrando as principais funcionalidades e análise dos resultados obtidos.

A ferramenta provê boa usabilidade e fácil utilização. Por mais limitações que a versão de entrada possui, foi possível visualizar conceitos de privacidade na prática e simular o registro das preferências de utilização dos dados do usuário através de um modelo genérico proposto por Werner (2017).

A familiarização das ferramentas para edição de políticas de privacidade está em ascensão nos tempos atuais. As legislações no mundo todo apontam para uma direção única, de criar uma cultura de respeito à privacidade dos dados dos usuários, como é o caso da LGPD (Lei Geral de Proteção de Dados) aqui no Brasil (ANDRADE, 2019) e RGPD (Regulamento Geral sobre Proteção de Dados) da União Europeia (UE), por exemplo.

Pelo o exposto, conceitos como Privacy by Design (PbD) surgem como tendência no desenvolvimento de software. O PbD exige incorporar a privacidade ao design de tecnologias nos estágios iniciais do processo de desenvolvimento (HADAR et. al. 2018) e evitar violações de dados e seus conseqüentes, sendo assim de natureza preventiva (DESIGN, 2012).

Van Der Sype e Maalej (2014) reconhecem a influência e potencial que os desenvolvedores possuem na contribuição de decisões favoráveis à privacidade do usuário nos estágios iniciais de desenvolvimento.

O presente trabalho trouxe os conceitos de privacidade e as tecnologias atuais para tal propósito e contribuiu de forma didática para que pessoas que não tenham pleno conhecimento em administração de redes possam modelar suas preferências em ferramenta de fácil uso.

6 - CONCLUSÃO

Neste trabalho foi apresentado um modelo genérico de definições de privacidade que foi implementado em ferramenta de gerenciamento de políticas de privacidade.

Os estudos bibliográficos que antecederam as implementações deram mais embasamento e maior entendimento sobre o problema. Foram apresentadas tecnologias como o ABAC, um modelo para controle de acesso em ascensão e a linguagem XACML. Pode-se observar também a motivação atual que permeia a privacidade e as novas legislações vigentes que tornam a sociedade civil mais justa, democrática e empoderada.

Diante das ferramentas disponíveis para edição e simulação de políticas XACML, destaca-se a SPT pelo fato de ser gratuita, acesso rápido e fácil para download, ter bom conteúdo de documentação na web e possuir interface intuitiva para seu uso. As demais ferramentas no mercado também conseguem entregar o mesmo resultado porém deixam a desejar na usabilidade e gastos em recursos e tempo (ref). O ACPT recebeu boa avaliação e também possui destaque visto a interface de criação, qualidade da ferramenta, automação de testes e geração automática do XACML. O SPT ainda sim chega a ser superior ao ACPT em alguns aspectos, como incluir todos os algoritmos de combinação do XACML 3.0, análise abrangente de verificação das políticas e grande cobertura nos testes em busca de falhas no controle de acesso.

Conclui-se, que a implementação mostrou como as definições funcionam na prática e que os resultados dos testes foram satisfatórios.

Para trabalhos futuros poderiam ser modelados em um gerenciador de políticas casos reais a fim de validar modelos mais específicos, sem a limitação de versões free, vários sujeitos e recursos e grandes casos de testes. Ademais, seria interessante o registro da integração do código XACML resultante com a camada de aplicação nas situações de controle de acesso. Outro estudo que poderia ser interessante é a apresentação detalhada de outras ferramentas como o WSO2, um gerenciador de identidades que demonstra, em análise superficial, ser completa, intuitiva e com suporte ao padrão XACML.

REFERÊNCIAS

Design, P. B. **Applying Privacy by Design Best Practices to SDG&E's Smart Pricing Program.** Toronto, Canadá, 2012. p. 5-6. Disponível em: <https://www.smartgridlegalnews.com/wp-content/uploads/sites/517/2012/03/PrivacyByDesign.pdf>Acessado em 05 de novembro de 2019.

HADAR, I. et al. **Privacy by Designers: Software Developers' Privacy Mindset.** 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE). Gothenburg, 2018. Disponível em: <https://ieeexplore.ieee.org/document/8453098>. Acessado em 04 de dezembro de 2019.

PATTAKOU, A. et al. **Towards the Design of Usable Privacy by Design Methodologies.** 2018 5th International Workshop on Evolving Security & Privacy Requirements Engineering. Canadá, 2018. p. 1-5. Disponível em: <https://www.computer.org/csdl/proceedings-article/espre/2018/842000a001/17D45WXIkli>. Acessado em 04 de dezembro de 2019.

Hadar I. *et al.*, **"[Journal First] Privacy by Designers: Software Developers' Privacy Mindset,"** ***2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE),*** Gothenburg, 2018, p. 260-268..Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453098&isnumber=8453044>Acessado em 05 de novembro de 2019.

Van Der Sype YS, Maalej W (2014) **On lawful disclosure of personal user data: what should app developers do?** 7th International Workshop on Requirements Engineering and Law (RELAW), IEEE 2014

SILVA, B. M**. Uma Proposta de Controle de Acesso como Serviço para Federação de Identidades.** Brasil, 2018. p. 24-32. Disponível em: <https://monografias.ufrn.br/jspui/bitstream/123456789/7685/1/PropostaControleAcesso_Silva_2018.pdf>Acessado em 02 de novembro de 2019.

ANDRADE, M. R. **LGPD Brasil: como se adequar à Lei Geral de Proteção de Dados Pessoais**, 2019. Blog Conta Azul. Disponível em: <https://blog.contaazul.com/lgpd-lei-geral-protecao-dados-pessoais>. Acesso em: 05 de novembro de 2019.

INFOBEYOUND TECHNOLOGY LLC. **Security Policy Tool User Manual**, 2017. Manual de usuário. Disponível em: <https://securitypolicytool.com/Content/files/Security-policy-tool-user-manual.pdf>. Acesso em: 02 de novembro de 2019.

LIMA, P. R. B. D. **SGPCA - Sistema Gerenciador de Políticas de Controle de Acesso**. Santa Maria, RS, Brasil, 2008. p. 32-48 Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/8050/PAULORICARDOLIMA.pdf?sequence=1&isAllowed=y>. Acessado em 02 de novembro de 2019.

ANDERSON, A. H. A comparison of two privacy policy languages: Epal and xacml. In: **Proceedings of the 3rd ACM Workshop on Secure Web Services.** New York, NY, USA: ACM, 2006. (SWS '06), p. 53–60. ISBN 1-59593-546-0

HWANG, J. et. al. ACPT: A Tool for Modeling and Verifying Access Control Policies. In: **2010 IEEE International Symposium on Policies for Distributed Systems and Networks**. Fairfax, VA, USA, 2010. p. 1-4. Disponível em: <https://ieeexplore.ieee.org/document/5629938/>. Acessado em 28 de junho de 2019.

CERETTA, R. N. et. al. **SISTEMA GERENCIADOR DE POLÍTICAS DE CONTROLE DE ACESSO**. XXVII Encontro nacional de engenharia de produção. Foz do Iguaçu, PR, Brasil. 2007. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2007_TR640476_8846.pdf>. Acessado em 28 de junho de 2019.

SADKI, S.; BAKKALI, H. E. An approach for privacy policies negotiation in mobile health-cloud environments. In: **Cloud Technologies and Applications (CloudTech), 2015 International Conference on**. [S.l.: s.n.], 2015. p. 1–6.

TOKTAR. E. et al. **Uma Arquitetura baseada em XML para Políticas RSVP**. Curitiba, 2015. C PPGIA, PUCPR. Disponível em: <https://secplab.ppgia.pucpr.br/files/papers/2005-1.pdf> Acessado em 22 de abril de 2019.

OASIS, eXtensible Access Control Markup Language - XACML v 1.0, Feb. 2003.

HU, V. C. et al. **SP 800-162 – Guide to attribute based access control (ABAC) definition and considerations.** [S.l.], 2014. v. 800, n. 162, 1–46 p.

ANDERSON, A. H. A comparison of two privacy policy languages: Epal and xacml. In: **Proceedings of the 3rd ACM Workshop on Secure Web Services.** New York, NY, USA: ACM, 2006. (SWS '06), p. 53–60. ISBN 1-59593-546-0.

MONT, M.C et al.. **A Systemic Approach to Automate Privacy Policy Enforcement in Enterprises**. 2006. Disponível em: <https://petsymposium.org/2006/preproc/preproc_07.pdf>Acessado em 05 de dezembro de 2019

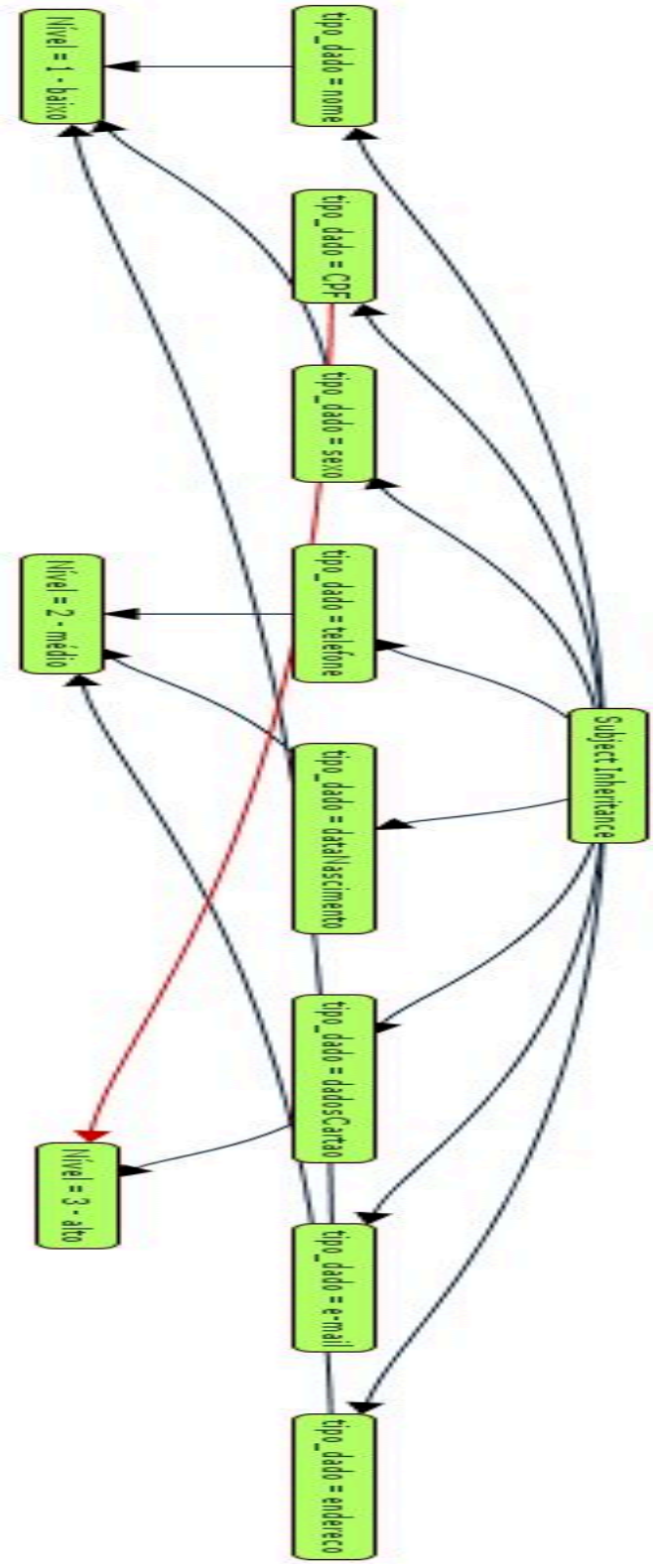
Ferraiolo, D. et al. **SP 800-178 - A comparison of Attribute Based Access Control (ABAC) standards for data service applications Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC)**. 2016. p. 5-12.

CHADWICK, D. W.; FATEMA, K. A privacy preserving authorisation system for the cloud. **Journal of Computer and System Sciences**, v. 78, n. 5, p. 1359 – 1373, 2012. ISSN 0022-0000. {JCSS} Special Issue: Cloud Computing 2011.

APÊNDICE A

Este apêndice apresenta a parte gráfica da herança da ferramenta SPT.

Figura 35 - representação gráfica herança definição 3



Fonte: Elaborada pelo autor

APÊNDICE B

Uso de ferramenta para edição e teste de políticas de gerenciamento de identidade

Gabriel Guimarães[1]

[1]Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476-88040-900 - Florianópolis – SC – Brasil

bielguimaraes94@gmail.com

Abstract. *With the increase in data generated, available service providers and new devices connected to the Internet, privacy needs to be addressed so that user information can be used securely. In this post, the present paper presents tools for building privacy policies, being less costly for the information data viewed by the broader stakeholders of a policy usually does not hold specific knowledge on inherent technologies. Testing in the tool provides more clarity on the practice of showing how the owner requests information about the collection, use, retention and disclosure of personal data, thus proving greater security on the web.*

Resumo. *Com o aumento de dados gerados, provedores de serviços disponíveis e novos dispositivos na conectados na internet se faz necessário tratar a privacidade para que as informações dos usuários estejam seguras. Isso posto, o presente trabalho apresentará ferramentas para que a construção de políticas de privacidade seja menos custosa aos donos da informação visto que os grandes interessados em uma política geralmente não detém o conhecimento específico em tecnologias inerentes. Os testes em ferramenta propiciarão mais clareza sobre a prática de expressar as preferências pelo dono da informação sobre a coleta, uso, retenção e divulgação de seus dados pessoais, provendo assim, maior segurança na web.*

1. Introdução

Com o aumento cada vez maior de aplicações web e computação em nuvem em geral, se faz necessário realizar controle de acesso às informações e gerenciar o volume de dados que circulam na rede. O gerenciamento de identidade trata como os dados relacionados a pessoas ou empresas que são usados para processar, autenticar e restringir os acessos na web (WERNER, 2017) A proposta neste trabalho é simular a execução de políticas de privacidade para o gerenciamento de identidade, partindo do pressuposto que o usuário deseja manter em sigilo, seguro e com acesso restringido de seus dados pessoais. As políticas auxiliam o usuário, dono da informação, por exemplo, para poder pré-definir quais atributos estarão disponíveis, para qual finalidade eles serão usados, por quanto tempo ou se deseja cifrar os seus atributos pessoais.

As políticas propostas foram pensadas e desenvolvidas para que qualquer usuário de uma cloud, inclusive leigos, consigam personalizar suas preferências de privacidade. Elas seriam apresentadas em uma aplicação na forma de perguntas, como o exemplo: “Você autoriza a disseminação de dados pessoais de acordo com o propósito específico? Qual?” ou “Você autoriza a disseminação de seus dados pessoais? Por quanto tempo?” (WERNER, 2017, p. 128 e 129). A propostas foram baseadas em 5 características essenciais em relação a privacidade dos dados dos usuários:

- Controle: quais atributos serão autorizados a acessar dados de algum servidor
- Retenção: por quanto tempo os dados estarão acessíveis à um provedor de serviços.
- Notificação: informa os usuários quando e de que forma seus dados foram processados
- Proteção: criptografia sobre os atributos
- Granularidade: regras de disseminação dos dados de forma minimizada, de acordo com o propósito

A simulação das propostas e conceitos apresentados serão feitos em ferramentas que auxiliam na modelagem de políticas de segurança na qual será analisado se o acesso à informação é garantido, quais atributos são usados, se as preferências do usuário no uso da informação está sendo respeitado, se a privacidade está sendo garantida e, se for o caso, apontar falhas de segurança nas propostas. Serão apresentadas as principais ferramentas e posteriormente a escolha da melhor para os testes e simulações.

Segundo Hadar (2018), citado por (Dinev e Hart 2006) e (Ayalon e Toch 2013), “Vários estudos mediram os riscos à privacidade e a tomada de decisões em domínios como comércio eletrônico e redes sociais on-line”. Esses estudos revelam que, na prática, os desenvolvedores estão dispostos a melhorar o nível de privacidade oferecido aos usuários obtendo uma melhor usabilidade do sistema. A necessidade que os conceitos de privacidade e ferramentas para gerenciamento das políticas de privacidade sejam divulgados é grande e o presente trabalho auxiliará em difundi-lo.

2. Políticas de privacidade

As discussões a respeito das políticas de privacidade em cloud tem tomado grandes proporções visto o grande crescimento dos usuários e dispositivos na web. Atualmente, os usuários na rede estão mais envolvidos na gestão de suas informações e preocupados na segurança dos mesmos. Segundo Anderson (2006, p. 1), “há uma preferência dos usuários em definir suas preferências de privacidade dando-lhes, assim, mais confiança e controle sobre seus dados pessoais.”

É importante que as políticas de privacidade descreva a maneira como a informação é manuseada, armazenada, usada e também deve permitir que o cliente tenha um controle máximo sobre os dados e processamento (SADKI; BAKKALI, 2015).

Com toda essa necessidade, a tendência é que cada vez mais empresas adotem os requisitos regulatórios atuais, como Sarbanes-Oxley, HIPAA e Diretiva da União

Européia sobre Privacidade de Dados para verificar sua conformidade com as políticas de privacidade (ANDERSON, 2006).

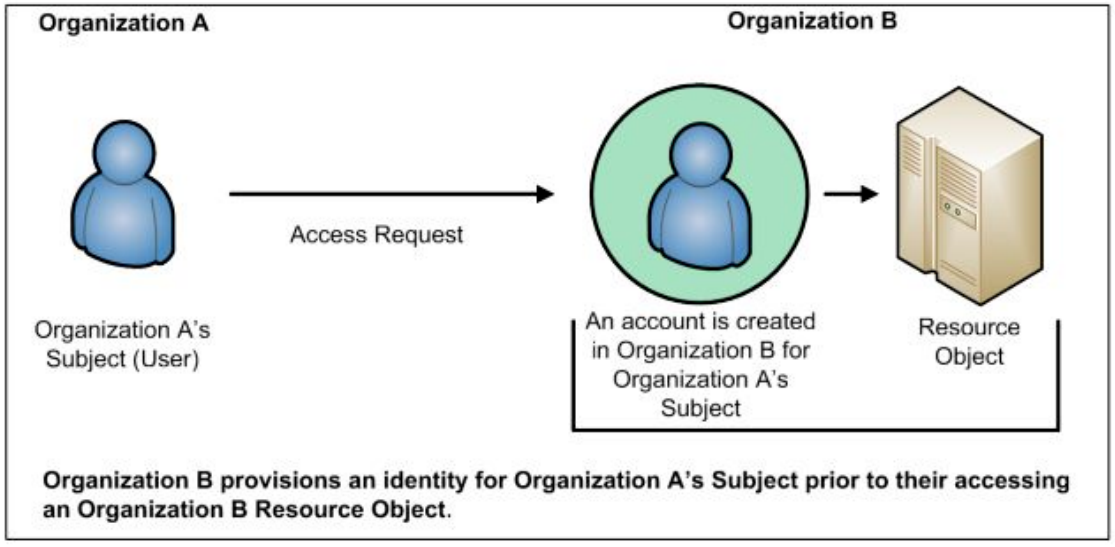
3. ABAC

Existem várias definições sobre o termo ABAC na literatura. Mas em resumo, ABAC seria um modelo de controle de acesso em que as permissões de execução são concedidas ou negadas a um determinado sujeito com base nos atributos de um objeto.

"Diante dos modelos de controle de acesso existentes, o ABAC (Controle de Acesso Baseado em Atributo) mostra-se ser o mais completo. Com ele é possível gerar políticas que expressam um conjunto de regras booleanas que pode avaliar muitos atributos diferentes" (HU, 2014, p. 5). Usando a estrutura da linguagem XACML, por exemplo, pode-se manipular regras e políticas, atributos (assunto, objeto, recurso, ação e ambiente) obrigações e conselhos. (HU, 2014, p. 5). Sua arquitetura é composta por políticas: Decisão de Política de Pontos (PDPs), Pontos de Cumprimento de Políticas (PEPs), Pontos de Administração de Políticas (PAPs) e Política de Pontos de Informação (PIPs) para controlar o acesso. (HU, 2014, p. 5). O ABAC é capaz de tomar decisões de controle de acesso a partir do conjunto dos atributos de um objeto, das condições de um ambiente ou conjunto de regras sem que elas sejam previamente especificadas diretamente a cada sujeito. Diante desses dados é possível criar políticas sem referência direta aos usuários que, podem ser muitos, permitindo gerá-las de forma mais genérica. (HU, 2014, p. 5)

Essa flexibilidade na criação dos controles de acesso é o que dá um grande benefício no uso do modelo ABAC. A figura 3 retrata um pedido de acesso entre organizações diferentes, que por sua vez possuem representações de identidades e políticas diferentes.

Figura 3 - requisição entre organizações



Fonte: Elaborada pelo autor

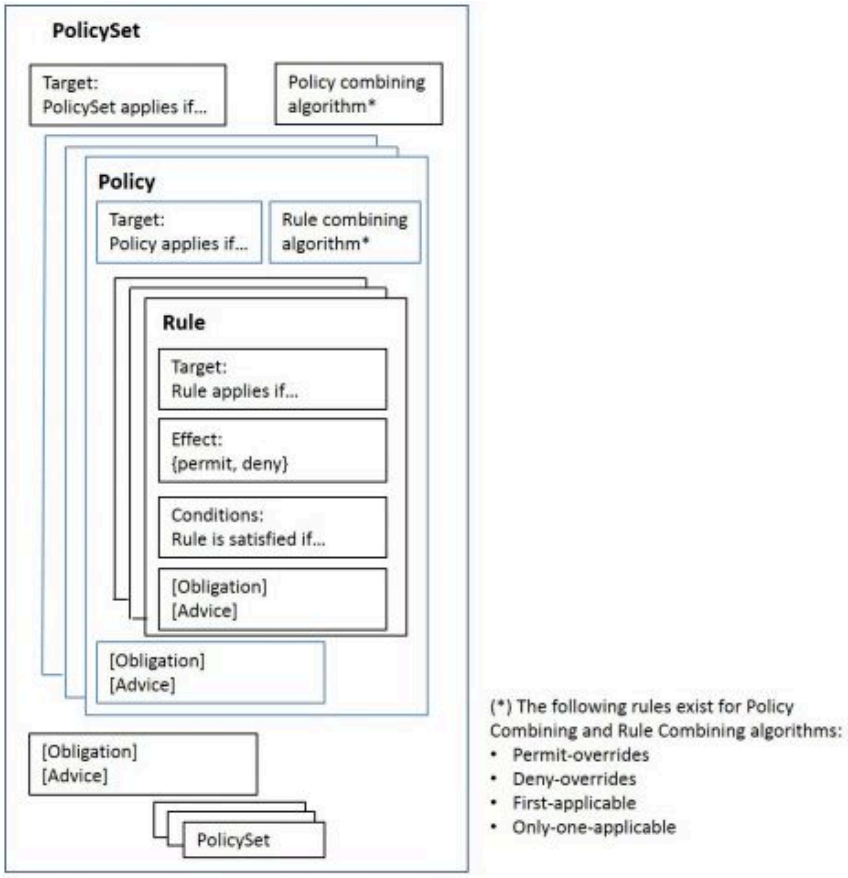
Do modo tradicional, a organização B exigiria que a identidade do solicitante fosse cadastrada e processada do lado de destino, com suas políticas e regras estabelecidas. (HU, 2014, p. 6). Com ABAC é possível tomar decisões de controle de acesso sem conhecimento prévio do objeto pelo sujeito ou conhecimento do assunto pelo proprietário do objeto e evita que a autorização seja atribuída antes da solicitação mas sim de forma dinâmica. (HU, 2014, p. 6).

4. XACML

XACML são iniciais formadas de "eXtensible Access Control Markup Language". Ela é uma linguagem de política de controle de acesso que obedece o padrão ABAC e foi desenvolvida pela organização de normas OASIS. Sua primeira versão foi publicada em 2003. XACML define um padrão de especificação que engloba sintaxe e semântica para representar solicitações, políticas, atributos e funções para computar decisões em recurso Ferraiolo (2016, p. 8) e também um modelo para representação das mensagens de requisição e resposta trocadas entre o PEP e o PDP (TOKTAR, 2016). Por se tratar de um padrão aberto, possuem pontos para extensão, possibilitando que um desenvolvedor defina novas funções, tipos de dados, combinações lógicas etc (LIMA, 2008).

Os atributos em XACML também seguem uma padronização. Segundo Lima (2008) toda requisição na linguagem conterá as categorias de atributos do usuário que são sujeito (subject), recurso (resource), ação (action) e ambiente (environment) e são especificados como pares nome-valor, por exemplo Papel = "médico" ou Tempo = “12:11” Ferraiolo (2016, p. 8). A estrutura da linguagem XACML pode ser vista conforme a figura 4:

Figura 4 - estrutura XACML



Fonte: Elaborada pelo autor

As PolicySets são formadas por políticas e as políticas formadas por regras. A resposta de uma requisição têm como resultado uma decisão entre quatro possíveis: permitir, negar, não aplicável (a ação solicitada não é válida) e indeterminado (erro na interpretação ou falha de informação sobre algum atributo).

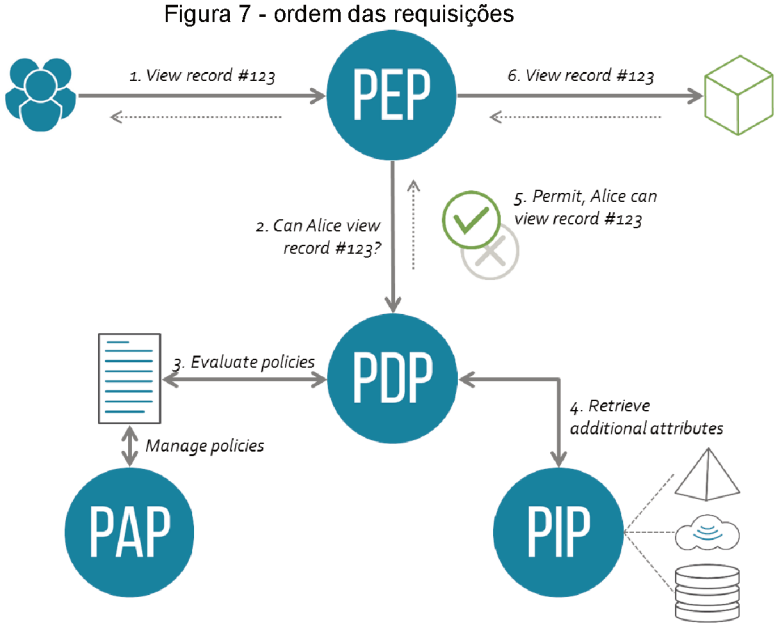
Como uma Política pode conter várias regras, e um PolicySet pode conter várias políticas ou PolicySets, cada regra, política ou PolicySet pode avaliar decisões diferentes (permitir, negar, não aplicável ou indeterminado). O XACML fornece uma maneira de unificar essas decisões. Esse processo é obtido por meio de uma coleção de algoritmos de combinação. Cada algoritmo representa uma maneira diferente de combinar múltiplas decisões locais em uma única decisão. Existem vários algoritmos de combinação definidos, segundo Ferraiolo (2016, p. 10):

- Deny-overrides: se qualquer decisão for avaliada como negar, ou se nenhuma decisão for avaliada como permitir, então o resultado é negar. Se todas as decisões forem avaliadas como permitir, o resultado será permitir.
- Permit-overrides: se qualquer decisão for avaliada como permissão, o resultado será permissão, caso contrário o resultado é negar.
- First-applicable: o resultado é o resultado da primeira decisão (permitir, negar ou indeterminado) quando avaliado em sua ordem listada.

- Only-one-applicable: se apenas uma decisão se aplica, o resultado é o resultado da decisão, e se mais de uma decisão se aplica, o resultado é Indeterminado.

Como mencionado, Toktar, 2016 diz que a linguagem XACML é poderosa para padronizar trocas de mensagem request e response entre PEP e PDP. Abaixo, tem-se a figura 7 ilustrando a sequência de passos de uma requisição.

Figura 7 - ordem das requisições



Fonte: Elaborada pelo autor

Quando um usuário deseja aplicar alguma ação sob um recurso, o PEP extrai os atributos e monta uma requisição em linguagem XACML Request até o PDP. O PDP é responsável por selecionar e aplicar as políticas definidas sobre os atributos da requisição, tomando a decisão de permitir ou negar a requisição em questão. Nele é selecionada a política junto ao PAP de acordo com a requisição. Informações dos atributos das entidades envolvidas, como sujeito, ambiente e recurso, são fornecidos pelo PIP. O PDP tendo tomado uma decisão, envia uma resposta XACML Response ao PEP. Por fim, o PEP aplica a decisão tomada pelo PDP, permitindo ou negando o acesso solicitado pelo sujeito (SILVA, 2018).

As linguagens de políticas estruturadas, como XACML, podem desempenhar um papel importante, principalmente se adotado um padrão universal de utilização. Enquanto muitas aplicações e plataformas têm seus próprios idiomas para controle de acesso, raramente são adequados para aplicar políticas de privacidade, e a falta de um padrão único torna a auditoria de conformidade um pesadelo. (ANDERSON, 2006, p. 1). A questão sobre adotar um padrão único na linguagem para políticas de privacidade é reforçada no artigo de Anderson (2006, p. 1), onde a linguagem usada para políticas de privacidade deve ser independente de plataforma e ser a mesma ou integrado à linguagem usada para políticas de controle de acesso, porque os dois tipos de políticas geralmente controlam o acesso aos mesmos recursos e não devem estar em conflito.

5. Ferramenta SPT

A ferramenta SPT (Security Policy Tool) oferece automatização e boa abrangência nos testes das políticas. Isso permite ter objetivos mais específicos no projeto, como identificar e corrigir possíveis erros nas políticas propostas e propor alterações se necessário. O site da ferramenta é bem completo. Dispõe de manual de usuário explicando todas as funcionalidades do software e exemplos de modelagem com documentação em PDF e vídeos explicativos com casos de testes, políticas no modelo ABAC e análise de resultados.

Para ter acesso ao software foi necessário criar um breve cadastro informando dados da instituição e título do trabalho de pesquisa no site oficial da ferramenta. Após isso o download é liberado. Na central de downloads do site encontra-se instaladores para Windows, Linux e MacOS. A versão mais completa custa 180 dólares.

6. Ferramenta SECTET-PL

O protótipo Sectet-PL utiliza um workflow baseado em UML (Unified Modeling Language) para gerar as políticas de controle de acesso em linguagem XACML. O administrador de rede faz a modelagem das permissões e a ferramenta se encarrega de traduzir e interpretar os dados para gerar o código para as políticas. Este gerador é destinado para a segurança de modelos de WorkFlow (CERETTA, 2007). O Sectet-PL foi projetado com o objetivo de gerenciar políticas de unidades hospitalares e não possui tratamento de conflitos.

7. Ferramenta ACPT

O ACPT (Access Control Policy Testing) foi desenvolvido por 4 pesquisadores em 2010, um deles membro da divisão de segurança computacional do NIST. Para se ter acesso a ferramenta não é algo tão fácil. É necessário fazer um breve cadastro no site do NIST (National Institute of Standards and Technology) e posteriormente solicitar à divisão de segurança computacional do NIST/ITL, via e-mail, a senha para descompactar o arquivo de download. Pode-se considerar algo bom pelo fato do acesso a ferramenta ser controlado, remete mais segurança e confiabilidade ao pesquisador usuário da ferramenta.

O ACPT, desenvolvida em Java, é composta por três funcionalidades principais. A primeira ajuda a especificar e combinar políticas com base em modelos de políticas existentes. A segunda, o ACPT analisa e converte uma política em formato executável, como XACML. Em terceiro lugar, para assegurar a correção das políticas, a ACPT conduz a verificação estática e dinâmica de uma política executando um conjunto de testes (HWANG, 2010) através da integração com outro software NuSMV. A verificação estática consiste em analisar com o auxílio de uma máquina de estados todas as possibilidades de resultados de uma política. Em caso de falha em algum teste a ferramenta não exibe sugestões de correção, cabe ao autor identificar e corrigir (HWANG, 2010).

O ACPT fornece interface gráfica para ajudar os autores de políticas a especificar políticas e suas propriedades de maneira interativa e eficaz. Além disso, o

ACPT também suporta recursos adicionais onde os autores de políticas podem editar, adicionar ou excluir políticas e seus atributos de forma interativa. A figura 17 mostra uma tela da ferramenta com suas funcionalidades:

8. Apresentação das definições

Na tese de Werner (2017, p. 129) temos a formalização das políticas apresentadas na tese distribuídas em 6 definições.

8.1. Definição 1

A primeira, é retratado o conceito de serviço. É definido por um possível conjunto de tipos de entrada, uma função que produz uma saída para cada entrada, e um conjunto de possíveis tipos de saída (WERNER, 2017). Em suma, todo serviço requisitado requer um conjunto de dados de entrada, na qual, serão processados para produzir um resultado. Um provedor de serviços exemplificado nessa mesma definição é a de transporte. O usuário fornece dados como nome e endereço, o provedor processa essa informação e retorna como saída um código de rastreio.

8.2. Definição 2

Para o nosso contexto, deve-se atentar sobre a privacidade, utilização e disseminação desses dados de entrada. A definição 2 aborda de forma mais objetiva esse contexto. Na tupla de itens contemplados estão contidos as seguintes informações:

- O tipo de dados para o qual a regra de política é definida (dT)
- conjunto de propósitos a se usar o dT (pR)
- tempo de conservação dos dados dT em horas (tM)
- o contexto especificado para determinado tipo de dados dT (cN)
- interesse na notificação sobre o uso dos dados (nT)
- interesse na cifragem dos dados (cP)
- interesse em obrigações a cumprir (oB)

A equação resultante da definição 2:

ProvedorServicos.RegrPrivacidade = {dT; pR; tM; cN; nT; cP; oB}

Para os atributos de notificação sobre o uso dos dados (nT) e cifragem dos dados (cP) os valores possíveis são “1” para sim ou “0” para não.

8.3. Definição 3

A definição 3 é retratado a preferência de privacidade do usuário para os dados de entrada. Um tipo de dado dT atrelado a um nível de sensibilidade de privacidade, bem como o propósito destinado. O nível pode ter valores de 1 a 3. Sendo 1 nível baixo, 2 médio e 3 nível alto. O detalhamento dos níveis é apresentado na seção 5.2.2 do artigo

de Werner, 2017 baseado em perfis de usuários e nos atributos que identificam diretamente uma pessoa. A equação é definida:

ProvedorServicos.PrefPrivacidade = [dT, Nivel, pR]

8.4. Definição 4

A definição 4 é propriamente a política de privacidade do usuário. São passadas 3 elementos:

- os dados, os atributos do usuário
- a regra de privacidade ao disseminar os dados ao serviço, seria a definição 2.
- preferência de privacidade sobre os dados de entrada, já vistas na definição 3.

A equação é definida:

ProvedorServicos.PoliticaPrivacidadeUsuario = Dados, RegrPrivacidade, PrefPrivacidade

As definições 5 e 6 tratam mais especificamente a solicitação e controle de acesso. Os atributos de suas equações já foram apresentadas pelas definições anteriores e seria redundante sua representação no contexto e objetivo do presente trabalho.

9. Modelagem da definição 2

A primeira modelagem feita na ferramenta foi a definição 2. O intuito é observar na prática com exemplificações das definições e fazer uma verificação do modelo com testes. No artigo de Werner, 2017 é retratado nesta definição um exemplo de regra de privacidade que diz que os tipos de dados (nome e CPF) podem serem usados apenas para fins de pagamento. O período de tempo para manter um valor de tipo nome e CPF é de 48 horas, apenas podem ser utilizados no contexto de compras online se o usuário deseja ser notificado sobre o uso dos dados, os dados devem ser transmitidos de forma cifrada e o sistema deve conferir o certificado de compra segura. A formalização do exemplo é representado:

SP.RegrP riv = [nome, cpf; pagamento; 48; compras; 1; 1; selo]

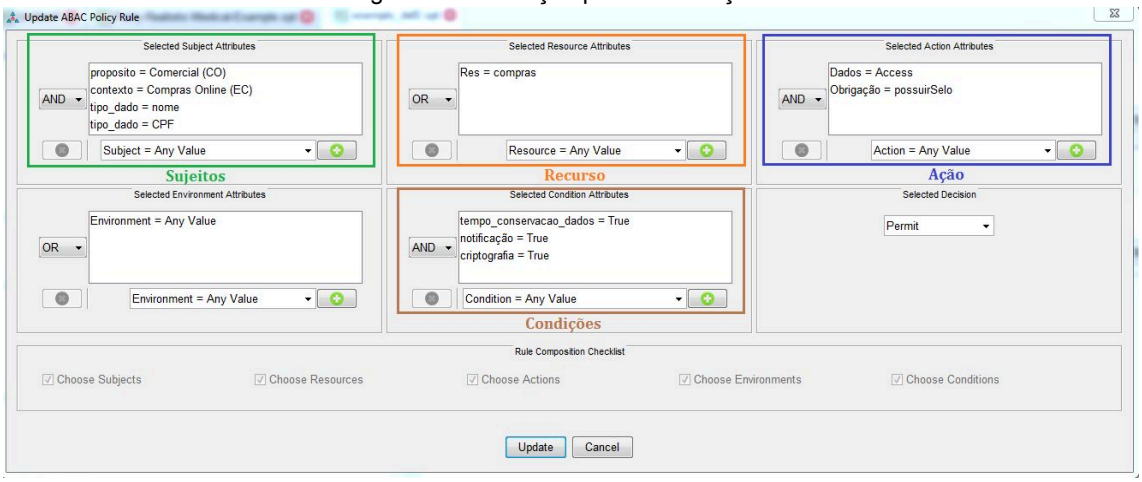
O atributo obrigações a cumprir (oB) foi modelado dentro de ações. A obrigação no exemplo refere-se ao provedor de serviços possuir um selo de compra segura.

Dentro de Condition (Condições), o atributo tempo de conservação dos dados, foi mapeado no SPT como um valor booleano. Sugere-se que se a requisição ainda está dentro do prazo definido pelo usuário, recebe valor true, senão, valor false. Mesma regra se aplica para notificação e criptografia. True caso a variável receba valor “1” e False caso receba valor “0”.

A figura 24 retrata a montagem da política ABAC da definição 2. Os sujeitos escolhidos foram propósito comercial, contexto de compra online (EC), tipo_dado,

nome e CPF. Recurso é compras. Ação é acessar o dado com a obrigatoriedade do selo de compra segura. Em relação as condições, o tempo_conservação_dados foi considerado True por estar dentro do valor de 48 horas, notificação e criptografia True. Após a implementação do exemplo pode-se iniciar os casos de testes.

Figura 24 - descrição política definição 2



Fonte: Elaborada pelo autor

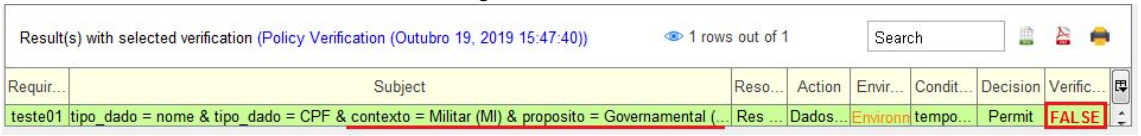
O teste01 foi considerado que um provedor de serviço gostaria de acessar os dados do usuário nome e CPF para um contexto militar e um propósito governamental. Tal situação é divergente com a política implementada no modelo ABAC e espera-se que o teste falhe. Da mesma forma, criou-se o teste2 com os valores de atributos corretos segundo a política ABAC da definição.

A funcionalidade Model Verification permite selecionar uma política, ou um conjunto delas, com um caso de teste a fim de analisar os resultados. Nessa funcionalidade da ferramenta, são realizadas todas as combinações possíveis para TRUE e FALSE. Alguns casos podem receber valor Not Applicable quando não há regra para um determinado recurso.

É comum nessa fase a percepção de conflitos entre regras. Caso ocorra, isso é possível resolver criando outras regras ou alterando os algoritmos de combinação do ABAC.

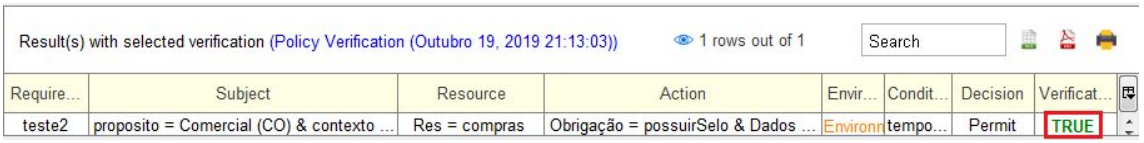
Abaixo temos as figuras 29 e 30 com o resultado do teste01 e teste2 respectivamente. Como era esperado, o teste01 falhou nos testes e recebeu FALSE. Já o teste2 recebeu TRUE por estar conformidade com a política.

Figura 29 - resultado teste 01



Fonte: Elaborada pelo autor

Figura 30 - resultado teste 2



Fonte: Elaborada pelo autor

9.1. Modelagem da definição 3

A modelagem da definição 3 teve algumas alterações em sua estrutura. O sujeito passou a ter propósito, o tipo_dado recebeu novos valores e foi acrescido os níveis de sensibilidade de privacidade como uma categoria de sujeito. Abaixo tem-se as figuras 31, 32 e 33 com valores para sujeito, recurso e ação.

Figura 31 - sujeito definição 3

Fonte: Elaborada pelo autor

Figura 32 - sujeito definição 3 continuação

Fonte: Elaborada pelo autor

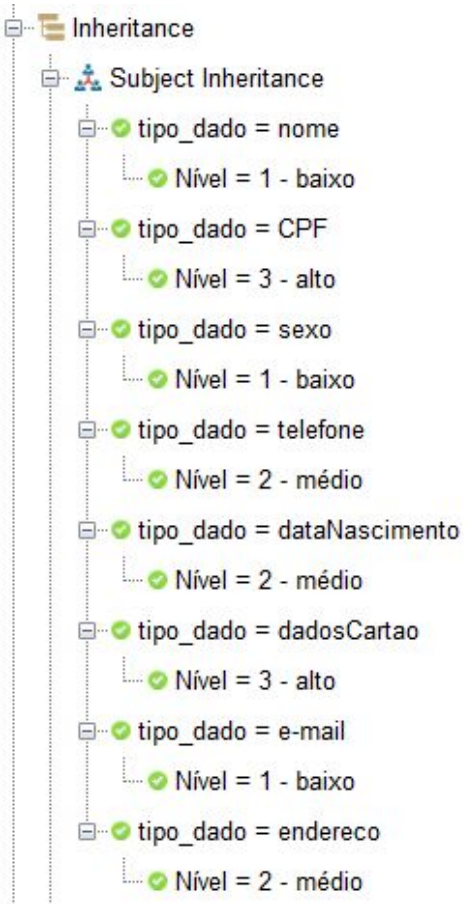
Figura 33 - recurso e ação definição 3

Fonte: Elaborada pelo autor

Dentro de Inheritance (herança), foi implementado uma pré-definição em relação níveis de privacidade para com cada tipo_dado do usuário (figura 34). O CPF, por exemplo, terá quando utilizado, nível 3 de privacidade. Em uma aplicação, os níveis poderiam ser mutáveis pelos usuários dependendo da profundidade da informação a ser acessada.

Na ferramenta SPT, a herança também pode ser construída graficamente conforme mostra a figura 35 do apêndice A. Os sujeitos beneficiados são ligados aos níveis que serão herdados.

Figura 34 - herança dos níveis definição 3



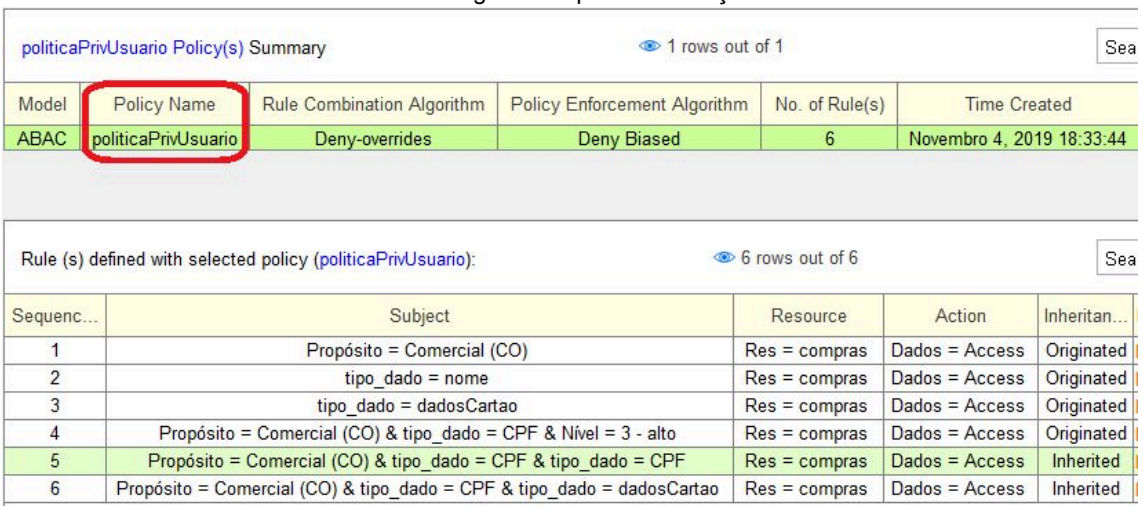
Fonte: Elaborada pelo autor

Isso posto, a definição 3 foi contida na ferramenta. O próximo passo, está descrito no subcapítulo seguinte com a modelagem da política de privacidade do usuário, a definição 4.

9.2 Modelagem da definição 4

A definição 4, já explicado no capítulo anterior, é composta por 3 elementos: dados de entrada, regra de privacidade (definição 2) e preferência de privacidade do usuário (definição 3). Tendo as definições 2 e 3 modeladas, pode-se agora escrever as políticas de uso dos atributos do usuário para um provedor de serviço. Na figura 38, temos a política 4 com suas regras:

Figura 38 - política definição 4



politicaPrivUsuario Policy(s) Summary — 1 rows out of 1

Model	Policy Name	Rule Combination Algorithm	Policy Enforcement Algorithm	No. of Rule(s)	Time Created
ABAC	politicaPrivUsuario	Deny-overrides	Deny Biased	6	Novembro 4, 2019 18:33:44

Rule (s) defined with selected policy (politicaPrivUsuario): — 6 rows out of 6

Sequenc...	Subject	Resource	Action	Inheritan...
1	Propósito = Comercial (CO)	Res = compras	Dados = Access	Originated
2	tipo_dado = nome	Res = compras	Dados = Access	Originated
3	tipo_dado = dadosCartao	Res = compras	Dados = Access	Originated
4	Propósito = Comercial (CO) & tipo_dado = CPF & Nível = 3 - alto	Res = compras	Dados = Access	Originated
5	Propósito = Comercial (CO) & tipo_dado = CPF & tipo_dado = CPF	Res = compras	Dados = Access	Inherited
6	Propósito = Comercial (CO) & tipo_dado = CPF & tipo_dado = dadosCartao	Res = compras	Dados = Access	Inherited

Fonte: Elaborada pelo autor

No topo da imagem temos o nome da política ABAC em destaque. Considera-se aqui que as regras de privacidade já foram configuradas pelo usuário para o serviço e que já estão incluídas na politicaPrivUsuario. O exemplo a seguir simboliza políticas e regras que o usuário destina o uso de seus dados bem como a preferência de privacidade para cada atributo. Pode-se observar que na linha de sequência número 1 o recurso compras só pode ser acessado em um propósito comercial. As linhas 2 e 3 diz que os tipos de dados permitidos são Nome e dados do cartão. A linha 4 é um pouco mais específica. O CPF, quando usado em propósito comercial, deve ser nível 3 de privacidade. Por esse motivo, as linhas 5 e 6 foram geradas por herança. Quando mencionado na regra anterior que o nível deve ser 3, automaticamente criou-se regras para outros tipo_dado de nível 3. No caso, CPF e dadosCartao. Em suma, é possível acessar o recurso compras apenas com nome ou dados do cartão. Porém quando usado CPF, só é possível com nível 3, seja por herança ou não. A figura 39 mostra que ao fornecer outro nível diferente do que 3 para propósito comercial, a requisição falha.

Figura 39 - resultado caso de teste nível errado

Result(s) with selected verification (Policy Verification (Novembro 4, 2019 19:31:01)) — 1 rows out of 1

Requirement Sch...	Subject	Resource	Action	Verification Result
nivelErrado	Propósito = Comercial (CO) & tipo_dado = CPF & Nível = 2 - médio	Res = compras	Dados = Access	FALSE

Fonte: Elaborada pelo autor

10. Análise dos resultados

A ferramenta provê boa usabilidade e fácil utilização. Por mais limitações que a versão de entrada possui, foi possível visualizar conceitos de privacidade na prática e simular o

registro das preferências de utilização dos dados do usuário através de um modelo genérico proposto por Werner (2017).

A familiarização das ferramentas para edição de políticas de privacidade está em ascensão nos tempos atuais. As legislações no mundo todo apontam para uma direção única, de criar uma cultura de respeito à privacidade dos dados dos usuários, como é o caso da LGPD (Lei Geral de Proteção de Dados) aqui no Brasil (ANDRADE, 2019) e RGPD (Regulamento Geral sobre Proteção de Dados) da União Europeia (UE), por exemplo.

Pelo o exposto, conceitos como Privacy by Design (PbD) surgem como tendência no desenvolvimento de software. O PbD exige incorporar a privacidade ao design de tecnologias nos estágios iniciais do processo de desenvolvimento (HADAR et. al. 2018) e evitar violações de dados e seus conseqüentes, sendo assim de natureza preventiva (DESIGN, 2012). Van Der Sype e Maalej (2014) reconhecem a influência e potencial que os desenvolvedores possuem na contribuição de decisões favoráveis à privacidade do usuário nos estágios iniciais de desenvolvimento.

O presente trabalho trouxe os conceitos de privacidade e as tecnologias atuais para tal propósito e contribuiu de forma didática para que pessoas que não tenham pleno conhecimento em administração de redes possam modelar suas preferências em ferramenta de fácil uso.

11. Conclusão

Neste trabalho foi apresentado um modelo genérico de definições de privacidade que foi implementado em ferramenta de gerenciamento de políticas de privacidade.

Os estudos bibliográficos que antecederam as implementações deram mais embasamento e maior entendimento sobre o problema. Foram apresentadas tecnologias como o ABAC, um modelo para controle de acesso em ascensão e a linguagem XACML. Pode-se observar também a motivação atual que permeia a privacidade e as novas legislações vigentes que tornam a sociedade civil mais justa, democrática e empoderada.

Diante das ferramentas disponíveis para edição e simulação de políticas XACML, destaca-se a SPT pelo fato de ser gratuita, acesso rápido e fácil para download, ter bom conteúdo de documentação na web e possuir interface intuitiva para seu uso. As demais ferramentas no mercado também conseguem entregar o mesmo resultado porém deixam a desejar na usabilidade e gastos em recursos e tempo (ref). O ACPT recebeu boa avaliação e também possui destaque visto a interface de criação, qualidade da ferramenta, automação de testes e geração automática do XACML. O SPT ainda sim chega a ser superior ao ACPT em alguns aspectos, como incluir todos os algoritmos de combinação do XACML 3.0, análise abrangente de verificação das políticas e grande cobertura nos testes em busca de falhas no controle de acesso.

Conclui-se, que a implementação mostrou como as definições funcionam na prática e que os resultados dos testes foram satisfatórios.

Para trabalhos futuros poderiam ser modelados em um gerenciador de políticas casos reais a fim de validar modelos mais específicos, sem a limitação de versões free, vários sujeitos e recursos e grandes casos de testes. Ademais, seria interessante o registro da integração do código XACML resultante com a camada de aplicação nas situações de controle de acesso. Outro estudo que poderia ser interessante é a apresentação detalhada de outras ferramentas como o WSO2, um gerenciador de identidades que demonstra, em análise superficial, ser completa, intuitiva e com suporte ao padrão XACML.

12. Referências

Design, P. B. Applying Privacy by Design Best Practices to SDG&E's Smart Pricing Program. Toronto, Canadá, 2012. p. 5-6. Disponível em: <https://www.smartgridlegalnews.com/wp-content/uploads/sites/517/2012/03/PrivacyByDesign.pdf>Acessado em 05 de novembro de 2019.

Hadar I. et al., "[Journal First] Privacy by Designers: Software Developers' Privacy Mindset," 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE), Gothenburg, 2018, p. 260-268..Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453098&isnumber=8453044>Acessado em 05 de novembro de 2019.

Van Der Sype YS, Maalej W (2014) On lawful disclosure of personal user data: what should app developers do? 7th International Workshop on Requirements Engineering and Law (RELAW), IEEE 2014

SILVA, B. M. Uma Proposta de Controle de Acesso como Serviço para Federação de Identidades. Brasil, 2018. p. 24-32. Disponível em: <https://monografias.ufrn.br/jspui/bitstream/123456789/7685/1/PropostaControleAcesso_Silva_2018.pdf>Acessado em 02 de novembro de 2019.

ANDRADE, M. R. LGPD Brasil: como se adequar à Lei Geral de Proteção de Dados Pessoais, 2019. Blog Conta Azul. Disponível em: <https://blog.contaazul.com/lgpd-lei-geral-protecao-dados-pessoais>. Acesso em: 05 de novembro de 2019.

INFOBEYOUND TECHNOLOGY LLC. Security Policy Tool User Manual, 2017. Manual de usuário. Disponível em: <https://securitypolicytool.com/Content/files/Security-policy-tool-user-manual.pdf>. Acesso em: 02 de novembro de 2019.

LIMA, P. R. B. D. SGPCA - Sistema Gerenciador de Políticas de Controle de Acesso. Santa Maria, RS, Brasil, 2008. p. 32-48 Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/8050/PAULORICARDOLIMA.pdf?sequence=1&isAllowed=y>. Acessado em 02 de novembro de 2019.

ANDERSON, A. H. A comparison of two privacy policy languages: Epal and xacml. In: Proceedings of the 3rd ACM Workshop on Secure Web Services. New York, NY, USA: ACM, 2006. (SWS '06), p. 53–60. ISBN 1-59593-546-0

HWANG, J. et. al. ACPT: A Tool for Modeling and Verifying Access Control Policies. In: 2010 IEEE International Symposium on Policies for Distributed Systems and Networks. Fairfax, VA, USA, 2010. p. 1-4. Disponível em: <https://ieeexplore.ieee.org/document/5629938/>. Acessado em 28 de junho de 2019.

CERETTA, R. N. et. al. SISTEMA GERENCIADOR DE POLÍTICAS DE CONTROLE DE ACESSO. XXVII Encontro nacional de engenharia de produção. Foz do Iguaçu, PR, Brasil. 2007. Disponível em: <http://www.abepro.org.br/biblioteca/enegep2007_TR640476_8846.pdf>. Acessado em 28 de junho de 2019.

SADKI, S.; BAKKALI, H. E. An approach for privacy policies negotiation in mobile health-cloud environments. In: Cloud Technologies and Applications (CloudTech), 2015 International Conference on. [S.l.: s.n.], 2015. p. 1–6.

TOKTAR. E. et al. Uma Arquitetura baseada em XML para Políticas RSVP. Curitiba, 2015. C PPGIA, PUCPR. Disponível em: <https://secplab.ppgia.pucpr.br/files/papers/2005-1.pdf> Acessado em 22 de abril de 2019.

OASIS, eXtensible Access Control Markup Language - XACML v 1.0, Feb. 2003.

HU, V. C. et al. SP 800-162 – Guide to attribute based access control (ABAC) definition and considerations. [S.l.], 2014. v. 800, n. 162, 1–46 p.

ANDERSON, A. H. A comparison of two privacy policy languages: Epal and xacml. In: Proceedings of the 3rd ACM Workshop on Secure Web Services. New York, NY, USA: ACM, 2006. (SWS '06), p. 53–60. ISBN 1-59593-546-0.

Ferraiolo, D. et al. SP 800-178 - A comparison of Attribute Based Access Control (ABAC) standards for data service applications Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC). 2016. p. 5-12.

CHADWICK, D. W.; FATEMA, K. A privacy preserving authorisation system for the cloud. Journal of Computer and System Sciences, v. 78, n. 5, p. 1359 – 1373, 2012. ISSN 0022-0000. {JCSS} Special Issue: Cloud Computing 2011.