2011 IEEE International Symposium on Policies for Distributed Systems and Networks

Behavior-based Policies for preserving Confidentiality in PCE-based Multi-Domain MPLS networks

Molka Gharbaoui*, Francesco Paolucci*, Barbara Martini†, Filippo Cugini† and Piero Castoldi*
*Scuola Superiore Sant'Anna, Pisa, Italy
Email: m.gharbaoui@sssup.it
†CNIT, Pisa, Italy

***Abstract*—Inter-domain path computations under the responsibility of cooperative Path Computation Elements (PCEs) guarantee effective network resource utilization and provide a basic level of information confidentiality. However, malicious PCEs belonging to different domains might misbehave by sending sequences of bogus requests and taking advantage of their interdependence to discover confidential information. In this work, we propose the use of XACML policies in order to avoid malicious utilizations of PCEP procedures and preserve confidentiality across domains. Policies are based on the analysis of the behavior of PCEP peers and the possible correlations among requests from which they might get access to private information.**

***Keywords*-Policies, XACML, PCE, Confidentiality.**

I. Introduction

The provisioning of Quality of Service (QoS)-guaranteed applications has driven the introduction of Traffic Engineering (TE) solutions in Multi Protocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks. TE relies on constraint-based path computations, which essentially consist in finding a shortest path between a source and a destination node, subject to constraints such as reservable bandwidth, diversity and resource class affinity. However, when multiple domains are involved in a path computation, a number of significant issues arise. Above all, the need to preserve information confidentiality across domains prevents the open advertisement of detailed intra-domain network resources. This considerably complicates the constraint-based path computation and it may also affect the inter-domain TE performance in terms of overall network resource utilization. As a matter of fact, network operators do not currently implement inter-domain TE techniques and the provisioning of QoS-guaranteed applications across multiple domains is performed manually, often requiring several weeks, and it typically relies on sub-optimal solutions (i.e., intra-domain independent path computations).

To overcome this issue, the Internet Engineering Task Force (IETF) has proposed a set of inter-domain TE techniques defined under the umbrella of the Path Computation Element (PCE) Architecture [1]. Such techniques, rather than exploiting information advertisement, rely on distributed path computations, performed in a cooperative way by entities (i.e., the PCEs) belonging to different domains.

This solution guarantees effective network resource utilization [1] and potentially provides an adequate level of information confidentiality. Despite authentication and encryption on path segments (i.e., Path-Key [2]), such potential might be jeopardized by the possibility for a PCE belonging to a different domain to maliciously issue bogus or false computation requests, aiming at discovering relevant confidential information related to other domains e.g., details on intra-domain network resources (i.e., total available bandwidth), congested portions of the network, node architectural limitations and constraints, recovery schemes, the ability/inability to support advanced network services and QoS-guaranteed applications. Confidentiality issues are arisen in [3] while, to the best of our knowledge, no mechanisms have been devised to address such problems.

This work extends the architectural principles presented in [4] and proposes a set of policies adopted by a Behavior-based PCE Authorization Policy (BPAP) scheme that prevents malicious utilizations of the Path Computation Element Protocol (PCEP) procedures and preserves confidentiality across domains. A couple of research works exist on security issues in the Multi-Domain environment [5] [6]. However, they focus on access control regulating the resource provisioning, and do not take into consideration confidentiality issues as this work does. BPAP applies on a two-step authorization scheme for PCE architecture where a

This work was partially supported by the STRONGEST project.

978-0-7695-4330-7/11 $26.00 © 2011 IEEE
DOI 10.1109/POLICY.2011.15

sequence of possibly correlated path computation requests is analyzed following predetermined attack patterns over a given resource, to estimate a potential malicious behavior of Path Computation Clients (PCCs) and hence detect a potential confidentiality attack. Specifically, a set of eXtensible Access Control Markup Language (XACML) policies is presented, which, based on the risk estimation of an incoming request and on the previous behavior of the requester (i.e., requests previously sent by the same PCC), allows or denies a PCC to access to the path computation procedure. In addition, a filter might be applied on the path parameters included in the response and sent along with the result of the path computation for further information restrictions. The proposed approach offers a reasonable trade-off between the need to preserve confidential information and the need to effectively utilize network resources, while guaranteeing good scalability performances.

The rest of the paper is organized as following. Section 2 describes the PCE architecture and the confidentiality issues in inter-PCE communication. Section 3 introduces our Behavior-based PCE Authorization Policy (BPAP) scheme. Section 4 then depicts the authorization policy and details the adopted XACML policies. Finally, section 5 presents our conclusions.

II. Confidentiality in PCE Architecture

A. PCE Architecture Review

The PCE architecture relies on two functional components: the PCE and the PCC. The PCE, possibly implemented on a dedicated server, is responsible to perform constraint-based path computations requested by PCCs which are typically implemented on a Network Management System (NMS) or a network node. In the inter-domain scenario, a PCE may also behave as a PCC, requesting path computations to a PCE belonging to a different domain. Communication between PCC and PCE is guaranteed by the recently standardized PCE Communication Protocol (PCEP) [3]. To perform path computations, PCC and PCE first open a PCEP session within a TCP session. A path computation request is then included within a PCReq message specifying all the requested parameters and constraints. Reply (i.e. PCRep message) is provided by the PCE specifying either the positive result (i.e., explicit path route) or negative result (i.e., no path found). Additional messages are also defined to close the PCEP session and to handle specific events and communication errors (e.g. Error (PCErr) and Notification (PCNtf) messages).

To perform inter-domain path computations, the PCE architecture defines two main procedures: the PCE-based Per-Domain (PPD) [7] and the Backward Recursive PCE-based Computation (BRPC) [8]. They both exploit a backward recursive technique. The path computation request is first forwarded between PCEs, domain-by-domain, until the PCE responsible for the domain containing the destination node is reached. The PCE in the destination domain then computes either a single sub-path (as in PPD) or a tree of virtual sub-paths (as in BRPC) to the destination. The result is passed back to the previous PCE which in turn expands the sub-path(s) and passes the result back until the source domain PCE completes the entire path computation. In the case of BRPC, the source PCE also selects the shortest path among those included in the final tree.

The main path computation parameters defined in [1] consider end-points (i.e., source and destination), connection bi-directionality and requested bandwidth. Other important parameters include: diversity (i.e., link, node and/or Shared Risk Link Group (SRLG) disjointness), the need for local protection (i.e., Fast ReRoute) and the application of BRPC procedure. In addition, PCEP specifications allow to provide information about failures in the path computation (i.e., NO-PATH information), to specify strict/loose sequences of hops to traverse or avoid, computed metric values, priority in the path computation, and information to perform re-optimization.

Hereafter, the main PCEP parameters are discussed highlighting their potential risk for a malicious utilization to break confidentiality:

1) Bandwidth (MPLS): Path computations requiring small values of bandwidth do not usually induce confidentiality issues. On the other hand, a request for a significant amount of bandwidth should require some careful treatment since it might allow the discovery of bottlenecks e.g., in case of negative reply.

2) Diversity and Bi-directionality: Path computation diversity (e.g., SRLG disjointness), local protection and bi-directionality imply the need to identify, within the requested domain, available resources along multiple disjoint routes or directions. This might aggravate the risk for discovering bottlenecks, topological limitations or node architectural constraints, in particular when associated to requests with relatively high bandwidth values.

3) Metrics: Metric values returned to a PCC might be used to infer intra-domain topological information. For example, subsequent identical requests for which the returned metric value changes, might indicate a variation in the intra-domain resource availability.

4) Backward Procedure: The backward nature of the PCEP procedures allows the requesting domain to retrieve information without providing any information about its own domain. This is particularly critical in the case of BRPC, where a tree between border nodes and the destination is returned together with the computed metric values.

B. Confidentiality in Inter-PCE communication

Inter-domain PCEP-based computations are performed upon general agreements between adjacent domains, which include technical (e.g., physical connectivity, interface switching capabilities) and economical specifications. The general agreement encompasses also confidentiality aspects, with the formalization of a set of rules and permissions aiming at defining the basic limitations in requests and replies due to confidentiality reasons. In particular, PCE and PCC will not exchange strict explicit list of traversed intra-domain hops and paths will be expressed in the form of an encoded Path-Key [2]. However, this basic level of trustworthiness agreement is not sufficient to fully guarantee the required level of confidentiality.

In [8], an overview of security considerations is provided. Requirements and possible solutions are indicated to address vulnerability aspects including spoofing (i.e., PCC or PCE impersonation), snooping (i.e., message interception), falsification, and denial of service. With reference to confidentiality aspects, [8] identifies the need to additionally define network policies aiming at preserving network information from bogus computation requests. Indeed, differently from connection requests triggered during signaling, PCEP-based computations do not imply the subsequent setup of the required connection, thus potentially enabling a malicious utilization of the PCE architecture. Also the time period between a positive reply and the related connection setup or timeout should be carefully treated, since a burst of requests could take place without being eventually concluded by any setup. Moreover, correlations among different path computation requests including PCEP parameters might introduce additional risks of breaking confidentiality. For example, multiple apparently independent path computation requests, targeting destinations located in the same geographical area, might hide a confidentiality attack. In particular, positive replies under certain constraints and negative replies under different constraints (e.g., link disjointness and SRLG disjointness) or in different time periods, could practically reveal lack/availability of intra-domain resources or intra-domain network performance (i.e., metrics).

Among the aforementioned events, and according to the signature-based detection approach [9], patterns, i.e sequences of requests with parameters matching specific criteria or with periodical behavior, may clearly reveal the possibility to be under attack [5]. For example, a sequence of requests targeting the same destination node and presenting values of bandwidth following periodical incremental step behavior may represent a possible attack candidate sequence to be classified under the suspected patterns. In this case it is likely that the client is periodically monitoring the resources availability towards a network node.

III. The Behavior-based PCE Authorization Scheme

A. Architecture

The architecture enables a BPAP scheme in the context of the PCE architecture that is shown in Fig. 1. It refers to a single domain which cooperates with adjacent domains to perform inter-domain path computations. As in [10], one or more PCCs per adjacent domain as well as one or more PCEs within the reference domain are considered. A PCE is equipped with a PCEP interface to handle PCEP communication and with a Path Computation Solver (PCS) to perform path computations. The architecture follows the approach based on a Policy Enforcement Point (PEP) and a Policy Decision Point (PDP) [11], encompassing two additional elements: a new PCE add-on component: the Authorization Policy Enforcement Controller (APEC) and a centralized Authorization Policy Server (APS). APEC plays the role of PEP, while the BPAP module acts as PDP.

APEC is introduced to filter the inter-domain path computation requests and replies. Concerning input requests, APEC performs basic authorization evaluations through simple permit/deny conditions specified in the form of access lists. Path computation results are also parsed by APEC and specific output information filtering is performed, based on local confidentiality rules.

APS is introduced to run, when needed, more sophisticated authorization policies based on a detailed analysis of the behavior of the PCC with regard to past path computation requests and the risk to confidentiality related to incoming and previous requests. To accomplish the latter task, APS utilizes a specific BPAP module and resorts to a Request Database (RDB) per domain. RDB stores all the details of the completed path computations handled by APS for each requesting domain, limited within a reasonable period of time (e.g., six months).

Critical requests are tagged with a status, based on the PCRep outcome and the possible related subsequent setup event: 1) failure: the requested path computation failed and No-Path object was included within PCRep; 2) setup: successful path computation with ERO included within PCRep, followed by the related LSP setup procedure (i.e., signaling), 3) expired: ERO included within PCRep, not followed by the related signaling, with setup timeout expired (typical value 10 minutes); 4) pending: ERO included within PCRep, not followed by the related signaling, with setup timeout not expired yet. The request status is dynamically updated based on the path computation outcome and the events occurring after path computation. In particular, the pending state eventually changes into either

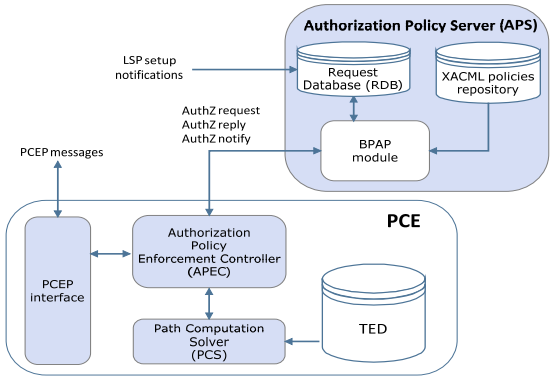

Fig. 1. BPAP two-step architecture: PCE and central APS.

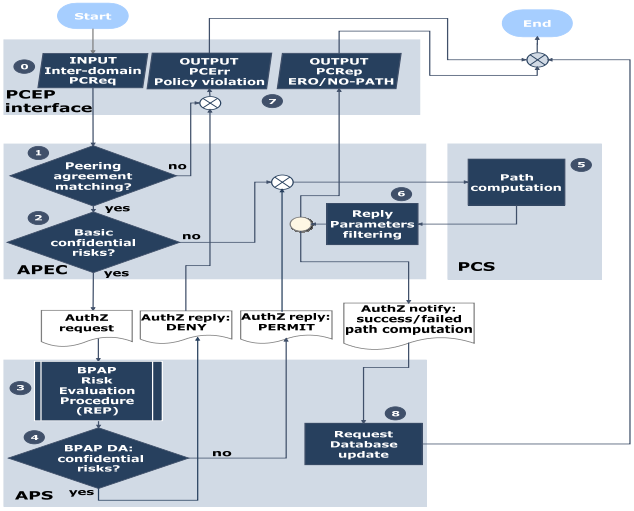

Fig. 2. PCE-APS authorization procedure workflow.

expired or setup. Communication between APEC and APS is achieved through the exchange of specific authorization messages. This might be implemented in Security Assertion Markup Language (SAML) carried by Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP). To maintain RDB databases, APS is notified with information on the final status (i.e., setup or expired) of the computed path, e.g. through Simple Network Management Protocol (SNMP) notifications from the Network Management System (NMS). The decoupling of the authorization evaluation performed by APEC (involved in all inter-domain path computations) and possibly by APS (only when complex evaluation is required) is introduced to better address the scalability requirements of the overall authorization scheme.

B. Authorization Procedure

Fig. 2 shows the entire procedure performed upon the arrival of an inter-domain path computation request, forwarded from the PCEP interface to the local APEC:

- Step 1: APEC first evaluates the incoming request on the basis of the general trustworthiness agreement defined with the adjacent domain. APEC decides whether immediately reject the authorization request, tagged as *unacceptable* (e.g., excessive request burst, parameters not allowed, such as excessive bandwidth) or proceed with the next step. In the former case, the procedure goes to step 7.
- Step 2: APEC evaluates the request on the basis of a simple authorization policy. Requests not determining risks to confidentiality (e.g., negligible bandwidth requirement) are tagged as *risk-free* and then directly forwarded to PCS (Step 5), while the remaining are tagged as *critical* and passed to APS for more careful authorization evaluation.
- Step 3: BPAP module evaluates the risk in terms of confidentiality of the incoming request taking into account its constraints, its possible correlation with previous requests and pattern detection. Such operation is referred to as Risk Evaluation Procedure (REP).
- Step 4: BPAP module decides to authorize or deny the request based on parameters computed at the previous step, through a Decision Algorithm (DA). APS replies to APEC with the following mutual exclusive options: a) Authorize the incoming request to be forwarded to the PCS for the subsequent path computation. The procedure continues at step 5. b) Deny the incoming request because of an excessive risk to confidentiality. The procedure continues at step 7.
- Step 5: PCS performs the required path computation and the result is passed to APEC.
- Step 6: APEC performs path computation reply filtering, based on the current confidentiality rules, and returns results to the PCEP interface. The procedure continues at step 7 and 8, performed in parallel.
- Step 7: The PCEP interface returns to the PCC the result of the path computation request in the form of: a) PCRep message with path computation failure (i.e., No-Path) or with the computed path. b) PCErr due to authorization failure.
- Step 8: Update of RDB through notification message from APEC to APS: the pending path computation request is now completed with PCRep information provided to the client and is inserted in the RDB, classified either *failure* state in case of No-Path or *pending* state in case of path computation

success.

If the requesting PCC receives a significant amount of authorization denials, the PCEP session between the domains may be eventually closed in order to verify and renegotiate the peering agreement. Warning messages could be defined to allow the requesting domain to become aware of its risky position.

C. Confidentiality Risk Evaluation and Decision Algorithm

A possible scenario of a Bandwidth monitor (Bm) attack consists in a PCEP client that sends a sequence of PCEP messages within a restricted period of time asking for LSPs towards the same target (i.e. destination address) and with required bandwidth values. Three suspicious sequences of requests have been identified: i) Bandwidth values that follow predefined patterns such as constant values, increasing-stepped values, decreasing-stepped values, and sawtooth values. ii) The periodical trend of the analyzed sequence i.e., one database entry is repeated N times. iii) The time sequence of the optional requested parameters occurrences (e.g., order of requests' arrivals, order of requested parameters)

The analysis consists in selecting the set of RDB entries coming from the same PCC and having the same target. The estimation of the probability to be under attack lies to the calculation of a risk parameter ρ that is bound to the monitoring of the resource through the correlations between the set of previously sent requests:

$$\rho = \alpha\rho_p + (1-\alpha)\rho_s \qquad (0 \leq \rho \leq 1)$$

where ρ_p accounts for the detection of at least one pattern, ρ_s is bound to the quantification of suspicious sequences of requests collected from the RDB and α is a (0,1) tunable weight that enhances or reduces the impact of the pattern discovery on the authorization decision.

We start by computing the ρ_p ($\in \{0,1\}$) that takes the maximum value 1 if there is a match with one of the predefined patterns. ρ_s is calculated taking into consideration the impact of the outcome of each request previously sent by the same PCC, as a weighted sum of the weight status of each entry in the portion extracted from the database:

$$\rho_s = \frac{N_{\mathcal{L}} - N_{\mathcal{L}}^{setup}}{N_{\mathcal{L}}^2} \sum_{\mathcal{L}} w_l \tag{1}$$

where $N_{\mathcal{L}}$ is the $\mathcal{L}$ set dimension, $N_{\mathcal{L}}^{exp}$ the number of *setup* tagged entries and w_l is the weight of the l-th entry, dependent on the status s_i

$$w_l = \begin{cases} 0 & s_l = setup \\ 0.5 & s_l = failure \\ 1 & s_l = expired \\ [0.5, 1) & s_l = pending \end{cases} \tag{2}$$

The w_l weights are introduced in order to tune the risk contribution given by each entry status. In particular, expired request entries are considered probable attack candidates, failure requests are considered intermediate potential attacks, pending requests worsen as the setup time increases, while setup requests are risk-free. ρ_S equals 0 if all the requests are setup (minimum alert state) and equals 1 if all requests are expired (maximum alert state). The parameter is designed also to smooth the alert state in case the PCC performs requests followed by setup.

A threshold T_A defines the authorization decision (i.e., a permit or a deny). If $\rho \leq T_A$, the request is authorized and is passed to the PCS. Otherwise, it is refused and a PCError message is sent back to the PCC.

IV. Authorization Policy

In order to preserve confidentiality and prevent the disclosure of private contents, a new set of agreements was defined to specify different classes of PCCs, i.e., profiles, respect to the set of information the PCC is allowed to receive in the path computation result, i.e., the PCEP Objects, including metrics, C flags, and type-length-values (TLVs) [3]. Three different profiles of disclosed information have been specified:

- Advanced (i.e. full set of information): Top category of PCCs where all details about failures are given back i.e., which object in the request caused the failure (i.e., No-Path) and what were the unsatisfied constraints. The metric may be returned if requested.

- Standard (i.e. restricted set of information): Some privileges are given to the PCCs of this class especially about the reasons that led to a No-Path response where explanations about its causes are given. In case of success, only the Path-Key is returned.

- Basic (i.e. minimal set of information): Encompasses authenticated PCCs coming from domains belonging to carriers with which no agreement was previously signed. In this case, no details are sent with the path computation result (No-Path without explanations or just a Path-Key) even if asked by the requester (e.g., metric).

Within each class of PCCs, three confidentiality levels were defined: Low, High and Critical, in order to consider the recent behaviors of a PCC during limited periods of time. According to the value of the risk parameter ρ, restrictions for confidentiality reasons might be applied on the PCEP objects sent in the

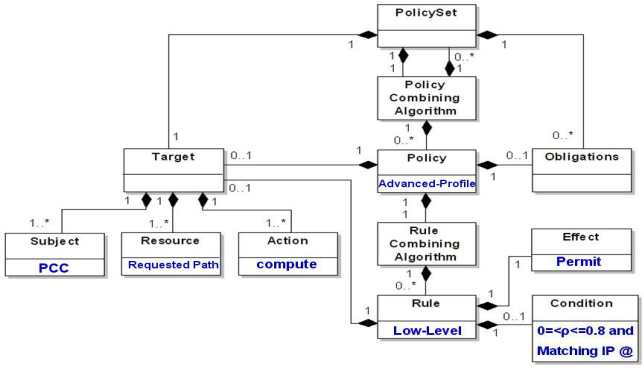

Fig. 3. XACML Policy Scheme [12].

response to a suspicious PCC (even if belonging to an advanced profile). Specifically:

- Low: The value of the risk parameter ρ is low enough to consider the PCC's request as safe. All agreed PCEP Objects can be described in the response.
- High: ρ increases and the behavior of the PCC starts to be suspicious. Some privileges are then removed.
- Critical: ρ is not anymore acceptable and its value is considered as critical. However, it is still lower than the threshold T_A and then some details are sent to the PCC. Once the threshold value exceeded, the session between the two peers is closed and a PCError Object is sent. In this case, the PCC is considered malicious.

Description of the XACML Policy

The set of policies regulating the APS decisions are expressed using the XACML language. XACML is an open standard XML-based language, designed to uniformly express policies and access rights, in an easy and understandable way [12]. It defines the rules needed to make the necessary authorization decisions and allows for the exchange of policy decisions between the PEP and the PDP. In this work, the policies are evaluated to determine whether or not the path computation request is allowed, i.e., does not affect or threatens the confidentiality of a receiving carrier. If accepted, and according to the calculated confidentiality level of the PCC, a filter might be applied on the information within the response sent back to the PCC, for further restrictions.

As shown in Fig. 3, a scheme for expressing the set of Policies, i.e., a PolicySet, contains a Target and a collection of Policies. The Target specifies the Subjects, Resources and Actions to identify the applicable policy statement and specify the set of requests to which it applies. The Target can also be found within the Policies and/or the Rules for expressing further restrictions. Each Policy contains a set of Rules combined in order to evaluate the outcome of the Policy, and a set of Obligations which are the actions performed by the PEP in conjunction with the enforcement of the authorization decision. A Rule may comprise a Description element and have a Condition section which further refines the applicability established by the Target. The result of Rules combination (i.e., Effect) may be a Permit, Deny, Not-Applicable (i.e., no Policy with matching target was found, or all Rules evaluate to false) or Indeterminate (i.e., an error occurred while evaluating a Rule).

In this work, when a request for authorization lands at the PEP, it sends it for evaluation to the PDP which sends back a response. The PDP arrives at a decision after evaluating the relevant policies and the rules within them. This evaluation is part of the BPAP Decision Algorithm.

The set of disclosed information profiles is mapped into a XACML PolicySet where the set of policies adopted by the BPAP module at the APS is shown in Fig. 4. It describes the different profiles of PCCs coupled with the defined confidentiality levels that aim at limiting the path parameters that are sent along with the result of the path computation, if permitted. For the sake of space, the policies were expressed in a compact way. Only the first level tags of the first policy (i.e., policy related to the advanced profile coupled with a low confidentiality level) were expanded. The `<Target>` element in the `<PolicySet>` defines the `<Resource>` to which the policy applies, which is in our case the Requested Path. The adopted set of policies signifies that a PCC can ask for a path computation only if it does not present any suspicious behavior and its IP address belongs to network area under the responsibility of an authenticated carrier. The `<PolicySet>` includes also a default `<Policy>` that is added in order to consider requests coming from unknown or not authenticated PCCs. Such PCCs are not allowed to ask for path computations and their requests are systematically denied.

Fig. 5 illustrates an example of a request for authorization sent from the PEP to the PDP. A request is a collection of attributes typically describing the requesting subject, the requested resource and the action that the subject wishes to perform on the resource. In our case, the `<Subject>` section contains the IP address of the PCC and its risk estimation calculated according to the arriving path-computation request and those previously sent and stored in the RDB. The `<Resource>` element indicates the requested resource (which is in our case a path with given parameters). The `<Action>` is to compute a path (only action supported by the policy).

Fig. 6 illustrates all the elements included in the first policy of the `<PolicySet>` which define the conditions applied to a PCC in order to launch a path computation. The `<Target>` element is empty which implies that the policy is restricted to the same target

```
<PolicySet PolicySetId="Behavior_Based_Policies"
      PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0
      :policy-combining-algorithm:permit-overrides">
-<Target>
-<Resources>
-<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0
      :resource-id" DataType="http://www.w3.org/2001/
      XMLSchema#string"/>
-<ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0
      :function:string-equal">
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema
      #string">Requested_Path</AttributeValue>
<ResourceAttributeDesignator DataType="http://www.w3.org
      /2001/XMLSchema#string" AttributeId="
      urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
</ResourceMatch>
</Resource>
</Resources>
</Target>
<!-- Policies 1/2/3: PCCs having an Advanced Profile -->
-<Policy PolicyId="Low-Advanced-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides">
+<Rule RuleId="Low_Level" Effect="Permit"></Rule>
-<Obligations>
+<Obligation ObligationId="Metric" FulfillOn="Permit"></
      Obligation>
+<Obligation ObligationId="C_Flag" FulfillOn="Permit"></
      Obligation>
+<Obligation ObligationId="TLV" FulfillOn="Permit"></
      Obligation>
</Obligations>
</Policy>
+<Policy PolicyId="High-Advanced-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
+<Policy PolicyId="Critical-Advanced-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
<!-- Policies 4/5/6: PCCs having a Standard Profile -->
+<Policy PolicyId="Low-Standard-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
+<Policy PolicyId="High-Standard-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
+<Policy PolicyId="Critical-Standard-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
<!-- Policies 7/8/9: PCCs having a Basic Profile -->
+<Policy PolicyId="Low-Basic-Profile" RuleCombiningAlgId="
      urn:oasis:names:tc:xacml:1.0:rule-combining-
      algorithm:permit-overrides"></Policy>
+<Policy PolicyId="High-Basic-Profile" RuleCombiningAlgId=
      "urn:oasis:names:tc:xacml:1.0:rule-combining-
      algorithm:permit-overrides"></Policy>
+<Policy PolicyId="Critical-Basic-Profile"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-
      combining-algorithm:permit-overrides"></Policy>
<!-- Policy 10: Unknown/Not authenticated PCCs -->
+<Policy PolicyId="Policy-by-default" RuleCombiningAlgId="
      urn:oasis:names:tc:xacml:1.0:rule-combining-
      algorithm:permit-overrides"></Policy>
</PolicySet>
```

Fig. 4. Set of policies adopted by BPAP at APS

specified in the PolicySet. In the <Rule> element, we mention that the action is to compute a path (<Action> element in the Target) and, through the <Condition> element, that the policy applies only to PCCs satisfying these two states: i) the PCC has an Advanced profile (based on its IP address), and ii) the risk parameter bound to this PCC belongs to a certain interval (e.g. [0, 0.8]). Finally, the <Obligations>

```
<Request>
<Subject>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:rho"
      DataType="http://www.w3.org/2001/XMLSchema#double">
<AttributeValue>0.3</AttributeValue>
</Attribute>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0
      :ip_address" DataType="http://www.w3.org/2001/XMLSchema
      #string">
<AttributeValue>2.2.2.5</AttributeValue>
</Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0
      :resource:resource-id" DataType="http://www.w3.org
      /2001/XMLSchema#string">
<AttributeValue>Requested_Path</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0
      :action:action-id" DataType="http://www.w3.org/2001/
      XMLSchema#string">
<AttributeValue>compute</AttributeValue>
</Attribute>
</Action>
<Environment/>
</Request>
```

Fig. 5. Policy Request

element specifies the set of actions to enforce along with the path computation. Both in the <PolicySet> and in the <Policy> elements, a PolicyCombiningAlgId attribute and a RuleCombiningAlgId, respectively, were defined. They specify a procedure for arriving at an authorization decision given the individual results of evaluation of a set of policies (respectively set of rules). In our case, we use the permit-overrides algorithm where if a single <Policy> or <Rule> element evaluates to Permit, then, regardless of the evaluation result of the other or <Policy> or <Rule> elements in the PolicySet (respectively in the Policy), the combined result is Permit.

Fig. 7 illustrates a policy response sent from the PDP to the PEP, as a result of evaluating the policy. The <Response> element encapsulates the authorization decision produced by the PDP . It includes a <Result> element for the requested resource (e.g. requested path) that contains the outcome of the decision request against the policy. The Permit information specifies that the PEP was allowed to perform the path computation. The <Status> element indicates whether errors occurred during the evaluation of the decision request, and optionally, information about those errors. In the <Obligations> section, the set of PCRep optional objects that must be sent along with the result of the computation are indicated. In this case (i.e. Advanced Agreement and low level risk), the C flag, the metric and TLVs are specified. Those obligations are actually implemented and executed in the PEP.

An experimental evaluation of the APS was carried out showing response times ranging from 100 to 150ms which demonstrate very good scalability performances. Time variations mainly depend on the number of requests already present in the RDB.

```
<Policy PolicyId="Low-Advanced-Profile" RuleCombiningAlgId="
      urn:oasis:names:tc:xacml:1.0:rule-combining-
      algorithm:permit-overrides">
<Target></Target>
<Rule RuleId="Low_Level" Effect="Permit">
<Target>
<Actions>
<Action>
<ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0
      :function:string-equal">
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#
      string">compute</AttributeValue>
<ActionAttributeDesignator DataType="http://www.w3.org/2001/
      XMLSchema#string" AttributeId="
      urn:oasis:names:tc:xacml:1.0:action:action-id"/>
</ActionMatch>
</Action>
</Actions>
</Target>
<Condition FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:and">
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:double-greater-than-or-equal">
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:double-one-and-only">
<SubjectAttributeDesignator AttributeId="
      urn:oasis:names:tc:xacml:1.0:rho" DataType="http://www.
      w3.org/2001/XMLSchema#double"/>
</Apply>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#
      double">0.0</AttributeValue>
</Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:double-less-than">
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:double-one-and-only">
<SubjectAttributeDesignator AttributeId="
      urn:oasis:names:tc:xacml:1.0:rho" DataType="http://www.
      w3.org/2001/XMLSchema#double"/>
</Apply>
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#
      double">0.8</AttributeValue>
</Apply>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:regexp-string-match">
<AttributeValue DataType="http://www.w3.org/2001/XMLSchema#
      string">2.2.2.*</AttributeValue>
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0
      :function:string-one-and-only">
<SubjectAttributeDesignator AttributeId="
      urn:oasis:names:tc:xacml:1.0:ip_address" DataType="
      http://www.w3.org/2001/XMLSchema#string"/>
</Apply>
</Apply>
</Condition>
</Rule>
<Obligations>
<Obligation ObligationId="Metric" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:Metric" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_Metric</AttributeAssignment>
</Obligation>
<Obligation ObligationId="C_Flag" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:C_Flag" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_C_Flag</AttributeAssignment>
</Obligation>
<Obligation ObligationId="TLV" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:TLV" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_TLV</AttributeAssignment>
</Obligation>
</Obligations>
</Policy>
```

Fig. 6. Policy related to Advanced Profile

V. Conclusion

In this paper, we have discussed the confidentiality issues in Multi-Domain PCE communications that take place when correlations between apparently independent path-computation requests are maliciously performed to access private contents. We proposed a BPAP scheme that analyzes these sequences of requests and decides to allow or deny access to path computations using XACML policies. Moreover, according to the profile of the requester PCC, an additional filter is applied on the response sent back along with

```
<Response>
<Result ResourceId="Requested_Path">
<Decision>Permit</Decision>
<Status><StatusCode Value="urn:oasis:names:tc:xacml:1.0
      :status:ok"/>
</Status>
<Obligations>
<Obligation ObligationId="C_Flag" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:C_Flag" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_C_Flag</AttributeAssignment>
</Obligation>
<Obligation ObligationId="TLV" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:TLV" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_TLV</AttributeAssignment>
</Obligation>
<Obligation ObligationId="Metric" FulfillOn="Permit">
<AttributeAssignment AttributeId="urn:oasis:names:tc:xacml:1
      .0:Metric" DataType="http://www.w3.org/2001/XMLSchema#
      string">Send_Metric</AttributeAssignment>
</Obligation>
</Obligations>
</Result>
</Response>
```

Fig. 7. Policy Response

the path computation result for further information restrictions. This automatic detection is considered as a first step to overcome the lack of confidentiality considerations in multi-domain PCE networks, and to allow for a reasonable trade-off between the need to protect information and the need to effectively utilize network resources, showing scalable performances in terms of response time.

References

[1] A. Farrel, J. P. Vasseur, and J. Ash, *"A Path Computation Element (PCE)-Based Architecture"*. IETF, RFC 4655, Aug 2006.

[2] R. Bradford, J.-P. Vasseur, and A. Farrel, *"Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism"*. IETF, RFC 5520, April 2009.

[3] J. Vasseur and J. Le Roux, *"Path Computation Element (PCE) Communication Protocol (PCEP)"*. IETF, RFC 5440, Mar 2009.

[4] F. Paolucci, F. Cugini, B. Martini, M. Gharbaoui, L. Valcarenghi, and P. Castoldi,, *"Preserving Confidentiality in PCEP-based Inter-domain Path Computation"*, in Optical Communication, 2010. ECOC 10. 36th European Conference on, Sept.2010, pp.12.

[5] Y. Demchenko, M. Cristea, and C. de Laat, *"XACML policy profile for multidomain network resource provisioning and supporting authorization infrastructure"*. in Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE International Symposium on, 2009, pp.98 101.

[6] S. Polito, M. Chamania, and A. Jukan, *"Extending the Inter-domain PCE Framework for Authentication and Authorization in GMPLS Networks"*. in Communications, 2009. ICC 09. IEEE International Conference on, 2009, pp.1 6.

[7] N. Bithar,R. Zhang, and K. Kumaki, *"Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)"*. IETF, RFC 5376, November 2008.

[8] J. Vasseur, R. Zhang, N. Bitar, and J. Le Roux, *"A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths"*. IETF, RFC 5441, Apr 2009.

[9] J. McHugh, A. Christie, and J. Allen, *"The Role of Intrusion Detection Systems"*. IEEE SOFTWARE, September/October 2000.

[10] A. Farrel and I. Bryskin, *"GMPLS: Architecture and Applications"*. Morgan Kauffmann, 2006.

[11] J. Vollbrecht, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, *"AAA Authorization Framework"*. IETF, RFC 2904, Aug 2000.

[12] OASIS, eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS, Feb. 2003.