Proposal: Domain Relationships

Prepared for W3C P3P 1.1 Working Group
Jack Humphrey (jhumphrey@coremetrics.com)
Revised December 2003

Introduction

As part of the P3P 1.1 effort, this document describes modifications to the P3P specification that would allow user agents to recognize when hosts in different domains are owned by the same entity or entities acting as agents for one another. These modifications would allow user agents to more intelligently apply privacy preferences, addressing implementation issues that have plagued many P3P deployments.[1]

This document contains an overview of the proposed modifications but not the specification-level details of the modifications, which will be provided in a subsequent document.

Limitations of Current Policy Reference File Syntax

Consider the web sites example.com and forinstance.com, which are owned by the same company and share some web site content, hosted on example.com. Some of that content includes "internal" banner ads that are generated by a servlet on example.com and promote certain features on example.com. Both sites are owned by the same company and wish to deploy a single P3P policy that describes their single corporate policy on data collection and usage.

A simple policy reference is deployed in the well-known location (/w3c/p3p.xml) on example.com. It looks like this:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
    <POLICY-REF about="/p3p/policy.xml#corporate">
      <INCLUDE>/*</INCLUDE>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

forinstance.com is configured to return the HTTP header

```
P3P: policyref="http://www.example.com/w3c/p3p.xml"
```

When a web browser visits a page on forinstance.com, the user agent applies the policy at the URI http://www.example.com/p3p/policy.xml#corporate. If an image on that page is

[1] http://www.w3.org/2002/p3p-ws/pp/coremetrics.pdf
"Agents and P3P"position paper presented to the W3C Workshop on the Future of P3P

served by example.com, the same policy would be applied to that image request, after looking up the policy reference file at the well-known location.

Since both the page request and the image request are covered by the same privacy policy, the user agent should be able to understand that any data collected (via cookies or otherwise) is being collected and used by the same entity, and therefore may decide that no extra privacy restrictions should be applied for the image request to the different domain.

If the hosts were flipped, and example.com was serving the page and forinstance.com the image, the same logic should hold with the exact same P3P deployment. This possibility brings up an interesting point. Third-party sites should not be able to "spoof" being covered by the first party privacy policy if they are not. If, in our example, forinstance.com was not part of the same company, and had different data collection and usage policies, this mechanism could allow it to claim to be covered by the example.com policy when in fact it is not. Without an additional mechanism in P3P, the user agent would not be capable of verifying the relationship and the burden would fall on example.com to be sure that third parties referenced in its site were referencing appropriate policies.

The same principle would apply if example.com, instead of being owned by the same entity as forinstance.com, were instead an agent of forinstance.com. Agent is defined by the P3P 1.0 specification as "a third party that processes data only on behalf of the service provider for the completion of the stated purposes."[2] The <ours> element of P3P does not distinguish between "ourselves" and "entities acting as our agents," and neither does this proposal.

Proposed Additions to Policy Reference Files

The proposed mechanism allows sites to declare hosts that are owned by the same entity or by agents in the policy reference file. These declarations would allow user agents to recognize and verify these relationships automatically.

Since forinstance.com references example.com's policy reference file, that file would have additional KNOWN-HOSTS sections:

```
<META xmlns="http://www.w3.org/2002/01/P3Pv1">
 <POLICY-REFERENCES>
    <POLICY-REF about="/p3p/policy.xml#corporate">
      <INCLUDE>/*</INCLUDE>
      <KNOWN-HOSTS>
        <HOST name="*.forinstance.com" entity-type="SAME"/>
      </KNOWN-HOSTS>
    </POLICY-REF>
 </POLICY-REFERENCES>
</META>
```

[2] P3P 1.0, section 3.3.5: <RECIPIENTS> element

This KNOWN-HOSTS section would allow example.com to declare hosts who may refer to the particular policy in the policy reference file. The KNOWN-HOSTS element could also be declared as a child element of POLICY-REFERENCES, in which case it would apply to all policies listed. The HOST name attribute denotes that forinstance.com hosts are allowed to refer to this policy reference file, and the entity-type attribute value "SAME" denotes that they are owned by the same entity. If entity-type were "AGENT" instead, it would denote that forinstance.com is an agent of example.com.

Browsers may cache the policy reference file based on an EXPIRY element in the policy reference file. The expiration information associated with that element should also be considered to apply to the known-hosts declarations; i.e. known host information may be cached along with the policy reference information.

Considerations for Compact Policies

Some user agents choose to only use compact policies to apply privacy preferences to cookies. Since the proposed mechanism relies on modifications to the policy reference file, user agents that rely solely on compact policies would not be able to verify these domain relationships.

Compact policies do not allow the same level of expressiveness as policy reference files. We propose a parallel mechanism using the HTTP P3P header to allow expression of "ours" type relationships with no ability to differentiate between same-entity and agent-of relationships.

P3P known-hosts Header

The P3P known-hosts header allows a host to specify a list of space-delimited hostname qualifiers that describe hosts owned by the same entity as the current host or agents of the current host. This list is equivalent to the list of known hosts in the policy reference file. In the example above, example.com would return the header:

```
P3P: known-host="*.forinstance.com"
```

and forinstance.com would return the header:

```
P3P: known-host="*.example.com"
```

When using these headers while evaluating compact policies, a user agent should only consider two hosts to have an "ours" relationship if each host has a matching known-host hostname qualifier for the other host. For simplicity of implementation, user agents should allow the trivial case of including the current host in the same-entity list.

Efficiency Concerns

For purposes of efficiency, hosts should not be required to return all hostname qualifiers for each of these new headers on every request. Instead they may tailor the header based on the request context, e.g. an entity owns 100 different domains, on hosts in forinstance.com, it may return only "*.example.com" if it can glean that the request was referred from example.com.

Implications for User Agents

Rules

To take advantage of the new expressiveness provided by the proposed modifications, user agents may implement the following high-level rules:

1. Hosts A and B should be considered to belong to the same entity if Host A refers to a policy reference file on host B, and that policy reference file contains a matching KNOWN-HOSTS entry for host B with type SAME.
2. Host B should be considered an agent collecting data solely on behalf of host A if Host A refers to a policy reference file on host B, and that policy reference file contains a KNOWN-HOSTS entry for host B with type AGENT.
3. Hosts A and B should be considered to have some kind of "our" relationship (either owned by the same entity or one is the agent of the other) if during compact policy evaluation, host A has a matching hostname qualifier in the P3P known-host header for the host B, and vice versa.

Cookie Playback

User agents should be aware that if they allow a cookie to be set based on a relationship established by known host declarations, they should verify that such a relationship exists at cookie playback time, and not send the cookie if it does not.