Discovering Provisioning Domain Names and Data

draft-ietf-intarea-provisioning-domains-01

P. Pfister, E. Vyncke, T. Pauly, D. Schinazi
Big News from IANA

<table>
<thead>
<tr>
<th></th>
<th>Option</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>IP Address/Prefix Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>18</td>
<td>New Router Prefix Information Option</td>
<td>[RFC4068]</td>
</tr>
<tr>
<td>19</td>
<td>Link-layer Address Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>20</td>
<td>Neighbor Advertisement Acknowledgement Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>21</td>
<td>PvD ID Router Advertisement Option (reclaimable in future)</td>
<td>[draft-ietf-intarea-provisioning-domains]</td>
</tr>
<tr>
<td>22</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>MAP Option</td>
<td>[RFC4140]</td>
</tr>
<tr>
<td>24</td>
<td>Route Information Option</td>
<td>[RFC4191]</td>
</tr>
<tr>
<td>25</td>
<td>Recursive DNS Server Option</td>
<td>[RFC5006][RFC8106]</td>
</tr>
<tr>
<td>26</td>
<td>RA Flags Extension Option</td>
<td>[RFC5175]</td>
</tr>
<tr>
<td>27</td>
<td>Handover Key Request Option</td>
<td>[RFC5269]</td>
</tr>
<tr>
<td>28</td>
<td>Handover Key Reply Option</td>
<td>[RFC5269]</td>
</tr>
</tbody>
</table>

-02 will include this number. Hackathon was done with this NDP Option Type
Changes in -01

• Remove all information about ‘metered’, ‘characteristics’
  • Still relevant but in another document?
• Clarify that PvD additional information is NOT to modify host stack behavior but only for applications
• Improve security & privacy sections
• Padding now to the 64-bit boundary
• Container approach to address a mix of PvD-aware and non PvD-aware hosts (see next slide)
PvD ID Option Format

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------------------
| Type     | Length | H | L | A | Reserved |
+-------------------------------------------
| Sequence Number                         |
+-------------------------------------------
|   PvD ID FQDN            ...                    |
+-------------------------------------------
|   Padding                  |
+-------------------------------------------
|   Router Advertisement message header  |
|   (Only present when A-flag is set)      |
+-------------------------------------------
|   Options ...                       |
+-------------------------------------------
```
### PvD ID Example

```
<table>
<thead>
<tr>
<th>Type: 21</th>
<th>Length: 12</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Sequence Number | 7 | e |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>a</td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>e</td>
<td>3</td>
</tr>
<tr>
<td>r</td>
<td>g</td>
<td>0</td>
</tr>
<tr>
<td>0 (padding)</td>
<td>0 (padding)</td>
<td>0 (padding)</td>
</tr>
</tbody>
</table>

RDNSS option (RFC 6106) length: 5
...
...
...

Prefix Information Option (RFC 4861) length: 4
...
...
...
### PvD ID Example

```
| Type: 21 | Length: 12 | 0|0|0| Reserved |
|-----------------|-----------|----------|
| Sequence Number | 7 | e |
| x | a | m | p |
| l | e | 3 | 0 |
| r | g | 0 | 0 (padding) |
| 0 (padding) | 0 (padding) | 0 (padding) | 0 (padding) |
| RDNSS option (RFC 6106) length: 5 |
| ... |
| ... |
| ... |
| Prefix Information Option (RFC 4861) length: 4 |
| ... |
| ... |
```
## PvD ID Example

<table>
<thead>
<tr>
<th>Type: 21</th>
<th>Length: 12</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Number</td>
<td>7e</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>a</td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>e</td>
<td>3</td>
</tr>
<tr>
<td>r</td>
<td>g</td>
<td>0</td>
</tr>
<tr>
<td>0 (padding)</td>
<td>0 (padding)</td>
<td>0 (padding)</td>
</tr>
</tbody>
</table>

**RDNSS option (RFC 6106) length: 5**

...  

**Prefix Information Option (RFC 4861) length: 4**

...
Implementation status

Linux - https://github.com/IPv6-mPvD

- pvdd: A Daemon to manage PvD IDs and Additional Data
- Linux Kernel patch for RA processing
- iproute tool patch to display PvD IDs
- Wireshark dissector
- RADVD and ODHCPD sending PvD ID

Implemented in one commercial vendor router
IPv6 mPvD + NEAT + SADR + Capport

= AWESOME
  Hack & Interop

Pierre Pfister
Eric Vyncke

Wenqin Shao

Kyle Larose

Michael Di Bartolomeo
This Hackathon: Complete test topology and interop.

https://github.com/IPv6-mPvD
Next steps

• Review is required

• Present the I-D to 6MAN & V6OPS WG
What about Security & Privacy
Confidentiality of Pvd Additional Information

• The well-known URL https://pvd-name.example.org/.well-known/pvd could contain some sensitive data (bandwidth, recursive DNS servers, ...)

• This well-known URL is guessable ;-)

• How to provide confidentiality ?

• 1) do not put anything which is really confidential

• 2) the HTTPS server should reject connections originated from prefixes not belonging to example.org
Spoofing the PvD ID

- Can an hostile party send rogue PvD, pretending to be example.org while they are hacker.org?
- No signature in the RA option (SeND not used)
Layer-2 Adjacent Attacker
Attackers are First Hop Router and PvD "Server"

PvD=good.com
Flag=H
PIO=2001:db8:bad::/64

H-flag is required
X.509 certificate is wrong
=> Do not trust
Attacker is the First Hop Router

PvD=good.com
Flag=H
PIO=2001:db8:bad::/64

{  
  name: "good.com";
  prefixes: ["2001:db8:beef:"];
}

H-flag is required
PIO not covered by "Prefixes"
=> Do not trust
Attacker is the First Hop Router with NPTv6

PvD=good.com
Flag=H
PIO=2001:db8:beef::/64

NPT
2001:db9:beef::
↔
2001:db8:bad::

H-flag is required
But cannot connect to the PvD server
=> Do not trust

My PvD are in 2001:db8:beef:: but this TLS client is in 2001:db8:bad::
=> Drop HTTPS request
Attacker Has a Foothold in "Good" PvD

IPv6 tunnel over foo

PvD=good.com  
Flag=H  
PIO=2001:db8:beef::/64

All appears good to host and PvD server... 
PvD approach does not help in this case 
But, it requires a foothold in good PvD
Host Privacy with Additional Information

• Each host will fetch the additional information on connection

• The HTTPS server will know the IP address of all clients and that the client is connecting...
  • Some privacy issues esp. if using EUI-64 or stable address

• Host can change to another IP address after fetching the file

• HTTPS belongs to the network operator (same as RADIUS, DHCP, ...)

• Anyway, it has more privacy than http://captive.example.com/hotspot-detect.html which belongs to another global operator