GNUnet for mesh communities

2016-05-04

BattleMesh v9, Porto

Daniel Golle <daniel@makrotopia.org>
Why bother?

Because a community mesh reality goes beyond wireless and routing algorithms

• Laggy (and costly) VPNs
Why bother?

Because a community mesh reality goes beyond wireless and routing algorithms

- Laggy (and costly) VPNs
- Evil firewalls/NAT
Why bother?

Because a community mesh reality goes beyond wireless and routing algorithms

- Laggy (and costly) VPNs
- Evil firewalls/NAT
- ‘just use 8.8.8.8’
Why bother?

Because a community mesh reality goes beyond wireless and routing algorithms

- Laggy (and costly) VPNs
- Evil firewalls/NAT
- ‘just use 8.8.8.8’
- ‘use Tor if you need privacy’
security and privacy
in contemporary community mesh networks

depends a lot on personal awareness and manual configuration
security and privacy
in contemporary community mesh networks

depends a lot on personal awareness and manual configuration

• macchanger
security and privacy in contemporary community mesh networks

depends a lot on personal awareness and manual configuration

• macchanger
• dnscrypt
security and privacy
in contemporary community mesh networks

depends a lot on personal awareness and manual configuration

• macchanger
• dnscrypt
• Tor/VPNs
no built-in security model in most mesh routing algorithms*!

*expectations: BMX7
security and privacy
in contemporary community mesh networks

no built-in security model
in most mesh routing algorithms*!

DoS or inserting malicious routes is trivial
security and privacy
in contemporary community mesh networks

Comparison
community mesh
vs.
commercial ISP
security and privacy in contemporary community mesh networks

Comparision community mesh vs. commercial ISP when accessing things on the web
security and privacy in contemporary community mesh networks

Comparision

when accessing things on the web*

*which is the most popular and sometimes only application of community mesh networks
## security and privacy in contemporary community mesh networks

<table>
<thead>
<tr>
<th>Typical community Mesh Network</th>
<th>Typical Commercial ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic routed through VPNs: (small set of) static source address(es) for all users of the mesh</td>
<td>Dynamic IPv4 address in a pool shared with tentousands of users, only ISP can map temporary addresses to users</td>
</tr>
<tr>
<td>Layer-2 MAC addresses and DHCP leases (containing hostnames and UIDs) may leave local administrative scope</td>
<td>Routing and NAT on Layer 3, MAC addresses and details about clients shouldn’t leave local realm (hopefully)</td>
</tr>
<tr>
<td>All other users may easily intercept or even maliciously re-route traffic</td>
<td>Central authority (ISP) governs routing, carriers involved on the way may intercept traffic (practically: any large TIER)</td>
</tr>
<tr>
<td>Informal hierarchies and knowledge-gap decide over priviledges</td>
<td>Cooperate hierachies and profit decide over priviledges</td>
</tr>
</tbody>
</table>

GNUnet for mesh communities - wbm v9
## security and privacy
### in contemporary community mesh networks

<table>
<thead>
<tr>
<th>Typical community Mesh Network</th>
<th>Typical Commercial ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic routed through VPNs: (small set of) static source address(es) for all users of the mesh (hundreds)</td>
<td>Dynamic IPv4 address in a pool shared with tenthousands of users, only ISP can map temporary addresses to users</td>
</tr>
<tr>
<td>Layer-2 MAC addresses and DHCP leases (containing hostnames and UIDs) may leave local administrative scope</td>
<td>Routing and NAT on Layer 3, MAC addresses and number of clients shouldn’t leave local realm (hopefully)</td>
</tr>
<tr>
<td>All other users may easily intercept or even maliciously re-route traffic</td>
<td>Central authority (ISP) governs routing, carriers involved on the way may intercept traffic (practically: any large TIER)</td>
</tr>
<tr>
<td>Informal hierarchies and knowledge-gap decide over priviledges</td>
<td>Cooperate hierachies and profit decide over priviledges</td>
</tr>
</tbody>
</table>
We are here
services inside mesh structures could (and maybe should) be implemented in a fundamentally different way than cloud (centralized) services
Architectural considerations

(users of) cloud services lack autonomy

default route failing

= users potentially able to communicate directly end up isolated
Architectural considerations

we need fault tolerance, graceful degradation and all those buzzwords the Erlang crowd has been preaching for over a decade...
Architectural considerations

X.509 (and thus TLS) is broken
what we need is some sort of distributed PKI
Architectural considerations

DNS is broken*
we need a decent distributed naming system

*DNSSec doesn’t help it, new TLDs also won’t help.
We need autonomous distributed applications

- to provide robust tools for self-organization
- to architecturally avoid all kinds of surveillance and censorship
- *endless list of pathetic arguments, democrazy, freedom-of-speech and all that*
We need autonomous distributed applications

what would an IoT light-switch you can trust have to look like?

Picture: Belkin WEMO Maker™
We need autonomous distributed applications

don't tell me you are going to rent your own server in a datacentre for a lightswitch…

or that you really believe that port-forwarding/UPnP, dyndns and ssh can beat them all and forever
We need
autonomous distributed applications

GNUnet or other secure P2P frameworks may be what you are looking for!
We need autonomous distributed applications

‘But P2P eats our bandwidth and gives us legal trouble, I don’t want that!’
We need autonomous distributed applications

‘most P2P tools didn’t work well in my mesh environment when I last tried (years ago)’
GNUnet

A general purpose modular P2P framework written in C.
• Lots of papers
GNUnet

- Lots of papers
- Some (mostly up to date) Documentation
• Lots of papers
• Some (mostly up to date) Documentation
• Lots of code :)
GNUnet goes embedded

OpenWrt port started in 2015 for wbm v8

- Focus on modularity
- mostly stateless / selective persistency
- UCI integration
- (basic) netifd integration
- (basic) firewall3 integration
GNUnet goes embedded

- core (~700kb) and 20+ modules packaged
- all transports and services work
- tunneling/VPN works
- Exit-to-ARPAnet setup works
- DNS-interception based integration of the GNUnet naming system works (still a bit tricky)
- sharing/updating, searching and downloading files/folders works
- Audio conversation maybe works :)

GNUnet for mesh communities - wbm v9
GNUnet future

• ‘social’ pub/sub API and CLI tool
• multi-user IRC-like chat based on PSYC working on top
• ‘consensus’ voting/contract system
• RESTful API
• More documentation
• Even further split things
• Testing! (volunteers needed)
Current injection-based wifi transport very slow due to missing rate-control

- Use Ad-Hoc, P2P or 11s interface instead, extract metrics from lower layers
GNUnet mesh future?

GNUnet has it’s own mesh-routing transport called ‘dv’
Online resources

- https://gnunet.org
- https://github.com/dangowrt/gnunet-15.05
- http://secushare.org/
GNUnet for mesh communities

2016-05-04

BattleMesh v9, Porto

Daniel Golle <daniel@makrotopia.org>