ipv6 - Task #7567

Add support for all TLS encapsulated protocols to IPv4-to-IPv6 incoming proxy

01/05/2020 04:00 PM - Moris Jones

Status:	Closed	**Start date:**	01/05/2020
Priority:	Normal	**Due date:**	
Assignee:	Nico Schottelius	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
PM Check date:			

Description

Here is a partial list of protocols

```
nntps           563/tcp snntp
ldaps           636/tcp
ldaps           636/udp
domain-s          853/tcp
domain-s          853/udp
ftps            990/tcp
telnets          992/tcp
pop3s            995/tcp
imaps            993/tcp
sip-tls         5061/tcp
sip-tls         5061/udp
mdns             5353/tcp
mdns             5353/udp
amqps            5671/tcp
customs          1001/tcp
customs          1001/udp
syslog-tls      6514/tcp
xmpp-server       5269/tcp jabber-server
xmpp-server       5269/udp jabber-server
amqps           5671/tcp
```

History

#1 - 01/05/2020 04:25 PM - Nico Schottelius

- Assignee changed from Nico Schottelius to Moris Jones

- Status changed from New to Feedback

Hey Moris,

I am not sure if this ticket is sensible, as it contains a lot of unused protocols. I like the other tickets more, which in my opinion make more sense to implement something that is actually in use.

I suggest to close it - ok with you?

#2 - 01/06/2020 11:48 AM - Moris Jones

The idea is that rather than implementing protocols one by one, implement all of them with a generic TLS proxy. Why be restrictive? IPv6 is about unlimited freedom, not putting people in a box. If a protocol is complicated like SMTP, drop it from the list. In such a case only implement it if it is important. Otherwise why not? How complicated can it be?

An excellent case here is RTP. When running a SIP or IAX exchange, hundreds or thousands of ports may be required. A typical usage is to assign port numbers 10,000 - 20,000 to this, but their is no standard.

It is annoying and offputting as a customer to have to request special treatment. Some won't bother and will just go elsewhere. And what if a customer wants a non-standard port usage, or is developing something new? Either you have to implement it especially for them or turn them away. What I propose is to give people the freedom to do whatever they want with their ports, the only limitation being that ports not assigned to a TLS encapsulated protocol will not work with the proxy, only on pure IPv6.

You want to market to geeks? Be generic, not bureaucratic. Get this proxy done, permanently, behind you and then forget about the idiots running the rest of the world and go do real stuff, like ucloud. Or ungleich linux. Or a floating datacenter.

#3 - 01/06/2020 11:51 AM - Moris Jones

- Status changed from Feedback to Waiting

- Assignee changed from Moris Jones to Nico Schottelius

#4 - 01/06/2020 12:12 PM - Moris Jones

There are other examples, such as running multiple sshd instances based on different authentication systems, each on a different port, or different web servers on different ports for different uses, or running a service on a non-standard port to stop script-kiddie attacks spamming the logfiles, or to hide from wrath while an exploit is being fixed. IPv4only isn't going away any time soon unfortunately and forcing people to do everything the 'standard' way with 'standard' tools is an approach that belongs in a shared hosting product, not with real private servers.

#5 - 01/06/2020 02:33 PM - Nico Schottelius

- Status changed from Waiting to Closed

Thanks a lot for creating this and the other tickets. I'll close this one in favor for the more specific services that we can actually implement. A generic/all protocol implementation will not be possible on application level, but only due to NAT64. And if we use 1:1 NAT64, nothing actually improves.