              Default Port for Internet Relay Chat (IRC) via TLS/SSL

Abstract

   This document describes the commonly accepted practice of listening
   on TCP port 6697 for incoming Internet Relay Chat (IRC) connections
   encrypted via TLS/SSL.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7194.

Table of Contents

1. Rationale

   Although system port assignments exist for IRC traffic that is plain
   text (TCP/UDP port 194) or TLS/SSL encrypted (TCP/UDP port 994)
   [IANALIST], it is common practice amongst IRC networks not to use
   them for reasons of convenience and general availability on systems
   where no root access is granted or desired.

   IRC networks have defaulted to listening on TCP port 6667 for plain
   text connections for a considerable time now. This is covered by the
   IRCU assignment of TCP/UDP ports 6665-6669.

   Similar consensus has been reached within the IRC community about
   listening on TCP port 6697 for incoming IRC connections encrypted via
   TLS/SSL [RFC5246].

2. Technical Details

2.1. Connection Establishment

   An IRC client connects to an IRC server. Immediately after that, a
   normal TLS/SSL handshake takes place. Once the TLS/SSL connection
   has been established, a normal IRC connection is established via the
   tunnel. Optionally, the IRC server may set a specific user mode
   (umode) for the client, marking it as using TLS/SSL. Again,
   optionally, an IRC server might offer the option to create channels
   in such a way that only clients connected via TLS/SSL may join.

   For details on how IRC works, see [RFC1459], [RFC2810], [RFC2811],
   [RFC2812], and [RFC2813]. Please note that IRC is extremely
   fragmented, and implementation details can vary wildly. Most
   implementations regard the latter RFCs as suggestions, not as
   binding.

2.2. Certificate Details

2.2.1. Server Certificate

   The IRC server's certificate should be issued by a commonly trusted
   certification authority (CA).

   The Common Name should match the Fully Qualified Domain Name (FQDN)
   of the IRC server or have appropriate wildcards, if applicable.

   The IRC client should verify the certificate.

2.2.2. Client Certificate

   If the client is using a certificate as well, it should be issued by
   a commonly trusted CA or a CA designated by the IRC network.

   The certificate's Common Name should match the main IRC nickname.

   If the network offers nick registration, this nick should be used.

   If the network offers grouped nicks, the main nick or account name
   should be used.

   If the network offers nick registration, the client certificate
   should be used to identify the user against the nick database. See
   [CERTFP] for a possible implementation.

3. Security Considerations

   The lack of a common, well-established listening port for IRC via
   TLS/SSL could lead to end users being unaware of their IRC network of
   choice supporting TLS/SSL. Thus, they might not use encryption even
   if they wanted to.

   It should be noted that this document merely describes client-to-
   server encryption. There are still other attack vectors like
   malicious administrators, compromised servers, insecure server-to-
   server communication, channels that do not enforce encryption for all
   channel members, malicious clients, or comprised client machines on
   which logs are stored.

   Those attacks can by their very nature not be addressed by client-to-
   server encryption. Additional safeguards are needed if a user fears
   any of the threats above.

   This document does not address server links as there are no commonly
   accepted ports or even back-end protocols. Ports and back-end
   protocols are normally established in a bilateral agreement. All
   operators are encouraged to use strong encryption for back-end
   traffic, no matter if they offer IRC via TLS/SSL to end users.

4. IANA Considerations

   An assignment of TCP port 6697 for IRC via TLS/SSL has been made.
   The service name is "ircs-u" and the description "Internet Relay Chat
   via TLS/SSL":

   ircs-u 6697/tcp   Internet Relay Chat via TLS/SSL

5. Normative References

   [RFC1459] Oikarinen, J. and D. Reed, "Internet Relay Chat Protocol",
              RFC 1459, May 1993.

   [RFC2810] Kalt, C., "Internet Relay Chat: Architecture", RFC 2810,
              April 2000.

   [RFC2811] Kalt, C., "Internet Relay Chat: Channel Management", RFC
              2811, April 2000.

   [RFC2812] Kalt, C., "Internet Relay Chat: Client Protocol", RFC
              2812, April 2000.

   [RFC2813] Kalt, C., "Internet Relay Chat: Server Protocol", RFC
              2813, April 2000.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246, August 2008.

6. Informative References

   [IANALIST] IANA, "Service Name and Transport Protocol Port Number
              Registry", <http://www.iana.org/assignments/
              service-names-port-numbers>.

   [TOP100]   netsplit.de, "IRC Networks - Top 100",
              <http://irc.netsplit.de/networks/top100.php>.

   [MAVERICK] netsplit.de, "IRC Networks - in alphabetical order",
              <http://irc.netsplit.de/networks/
              lists.php?query=maverick>.

   [CERTFP]   The Open and Free Technology Community, "OFTC -
              NickServ/CertFP",
              <http://www.oftc.net/oftc/NickServ/CertFP>.

7. Acknowledgements

   Thanks go to the IRC community at large for reaching a consensus.

   Special thanks go to the IRC operators who were eager to support port
   6697 on their respective networks.

   Special thanks also go to Nevil Brownlee and James Schaad for working
   on this document in their capacities as Independent Submissions
   Editor and Reviewer, respectively.

Appendix A. Supporting Data

   As of October 2010, out of the top twenty IRC networks [TOP100]
   [MAVERICK], ten support TLS/SSL. Only one of those networks does not
   support TLS/SSL via port 6697 and has no plans to support it. All
   others supported it already or are supporting it since being
   contacted by the author. A more detailed analysis is available but
   does not fit within the scope of this document.

Authors' Address

   Richard Hartmann
   Munich
   Germany

   EMail: richih.mailinglist@gmail.com
   URI: http://richardhartmann.de