

NAME

sc_warts2json — JSON dump of information contained in a warts file.

SYNOPSIS

sc_warts2json [*file* . . .]

DESCRIPTION

The **sc_warts2json** utility provides a JSON dump of information contained in a sequence of warts files. The output is the same as that which would have been provided by **scamper** if the JSON output option had been chosen instead of the warts output option when the data was collected.

EXAMPLES

The command:

```
sc_warts2json file1.warts file2.warts
```

will decode and print the contents of `file1.warts`, followed by the contents of `file2.warts`, to standard out.

The command:

```
gzcat file1.warts.gz | sc_warts2json
```

will print the contents of the uncompressed file supplied on stdin.

JSON FORMAT FOR TRACE

```
{
  "type": "trace",
  "version": "0.1",
  "userid": integer, the trace -U parameter
  "method": string, method passed with the -P parameter
  "src": string, source address
  "dst": string, destination address
  "rtr": string, the first-hop router passed with the -r parameter
  "sport": integer, source port for TCP/UDP traces
  "dport": integer, destination port for TCP/UDP traces
  "icmp_sum": integer, the ICMP checksum for ICMP traces
  "stop_reason": string, the reason for trace stopping
  "stop_data": integer, additional context for trace stopping
  "start": timestamp structure reporting when the trace began
  {
    "sec": integer, seconds since 1970
    "usec": integer, microseconds elapsed in second
    "ftime": string, formatted as YYYY-MM-DD HH:MM:SS
  }
  "hop_count": integer, the last TTL probed in this trace
  "attempts": integer, attempts per hop
  "hoplimit": integer, maximum TTL trace will probe
  "firsthop": integer, first TTL traceroute probed
  "wait": integer, seconds to wait before retrying a probe
  "wait_probe": integer, 10s of milliseconds to wait between probes
  "tos": integer, value used in ToS / DSCP field
  "probe_size": integer, size of probes to send
  "probe_count": integer, number of probes sent in total by this trace
  "hops": array of hop structures
```

```
[
{
  "addr": string, address received in response
  "name": string, PTR associated with address
  "probe_ttl": integer, TTL used for this probe
  "probe_id": integer, attempt number for this hop counted from 0
  "probe_size": integer, size of the probe sent
  "tx": timestamp structure reporting when the probe was sent
  {
    "sec": integer reporting seconds since 1970
    "usec": integer reporting microseconds elapsed in second
  }
  "rtt": integer.integer RTT in milliseconds with microsecond granularity
  "reply_ttl": integer, IP TTL value received in the response packet
  "reply_tos": integer, IP TOS value received in the response packet
  "reply_ipid": integer, IP ID value received in the response packet
  "reply_size": integer, size of the response
  "icmp_type": integer, type of ICMP response
  "icmp_code": integer, code of ICMP response
  "icmp_q_ttl": integer, IP TTL value in the quoted IP packet
  "icmp_q_ipid": integer, IP length value in the quoted IP packet
  "icmp_q_tos": integer, IP TOS value in the quoted IP packet
  "icmp_nhmtu": integer, MTU value if response was packet-too-big
  "tcp_flags": integer, TCP flags if response was TCP
  "icmpext": array of ICMP extension structures
  [
    {
      "ie_cn": integer, class number of ICMP extension
      "ie_ct": integer, class type of ICMP extension
      "ie_dl": integer, data length of ICMP extension
      "mpls_labels": array containing MPLS label stack entries
      [
        {
          "mpls_ttl": integer, the TTL in an MPLS ICMP extension
          "mpls_s": integer, the bottom-of-stack bit
          "mpls_exp": integer, the value in the experimental use field
          "mpls_label": integer, the 20-bit MPLS label
        }
      ]
    }
  ]
}
]
```

SEE ALSO

scamper(1), sc_wartsdump(1), sc_warts2text(1)

Ron Bonica, Der-Hwa Gan, Carlos Pignataro, and Dan Tappan, *ICMP Extensions for Multiprotocol Label Switching*, RFC 4950.

AUTHORS

sc_warts2json was written by Matthew Luckie <mjl@luckie.org.nz>. The JSON output formats for traceroute and ping were built on an initial prototype provided by Brian Hammond of Internap Network Services Corporation. The JSON output formats for dealias, tbit, and tracelb were designed and implemented by Matthew Luckie.