

NAME

netdump - protocol for transmitting kernel dumps to a remote server

SYNOPSIS

To compile netdump client support into the kernel, place the following lines in your kernel configuration file:

```
options INET
options DEBUGNET
options NETDUMP
```

DESCRIPTION

netdump is a UDP-based protocol for transmitting kernel dumps to a remote host. A netdump client is a panicking kernel, and a netdump server is a host running the **netdump** daemon, available in ports as *ports/ftp/netdumpd*. **netdump** clients are configured using the *dumpon(8)* utility or the **netdump** command in *ddb(4)*.

netdump client messages consist of a fixed-size header followed by a variable-sized payload. The header contains the message type, a sequence number, the offset of the payload data in the kernel dump, and the length of the payload data (not including the header). The message types are HERALD, FINISHED, KDH, VMCORE, and EKCD_KEY. **netdump** server messages have a fixed size and contain only the sequence number of the client message. These messages indicate that the server has successfully processed the client message with the corresponding sequence number. All client messages are acknowledged this way. Server messages are always sent to port 20024 of the client.

To initiate a **netdump**, the client sends a HERALD message to the server at port 20023. The client may include a relative path in its payload, in which case the **netdump** server should attempt to save the dump at that path relative to its configured dump directory. The server will acknowledge the HERALD using a random source port, and the client must send all subsequent messages to that port.

The KDH, VMCORE, and EKCD_KEY message payloads contain the kernel dump header, dump contents, and dump encryption key respectively. The offset in the message header should be treated as a seek offset in the corresponding file. There are no ordering requirements for these messages.

A **netdump** is completed by sending the FINISHED message to the server.

The following network drivers support netdump: *alc(4)*, *bge(4)*, *bnxt(4)*, *bxe(4)*, *cxgb(4)*, *em(4)*, *igb(4)*, *ix(4)*, *ixl(4)*, *mlx4en(4)*, *mlx5en(4)*, *re(4)*, *vtnet(4)*.

SYSCTL VARIABLES

The following variables are available as both `sysctl(8)` variables and `loader(8)` variables:

net.netdump.debug

Control debug message verbosity. Debug messages are disabled by default, but are useful when troubleshooting or when developing driver support.

net.netdump.path

Specify a path relative to the server's dump directory in which to store the dump. For example, if the **netdump** server is configured to store dumps in `/var/crash`, a path of "foo" will cause the server to attempt to store dumps from the client in `/var/crash/foo`. The server will not automatically create the relative directory.

net.netdump.polls

The client will poll the configured network interface while waiting for acknowledgements. This parameter controls the maximum number of poll attempts before giving up, which typically results in a re-transmit. Each poll attempt takes 0.5ms.

net.netdump.retries

The number of times the client will re-transmit a packet before aborting a dump due to a lack of acknowledgement. The default may be too small in environments with lots of packet loss.

net.netdump.arp_retries

The number of times the client will attempt to learn the MAC address of the configured gateway or server before giving up and aborting the dump.

SEE ALSO

`decryptcore(8)`, `dumpon(8)`, `savecore(8)`

HISTORY

netdump client support first appeared in FreeBSD 12.0.

BUGS

Only IPv4 is supported.

netdump may only be used after the kernel has panicked.