

**NAME**

**cap\_getaddrinfo**, **cap\_getnameinfo**, **cap\_gethostbyname**, **cap\_gethostbyname2**, **cap\_gethostbyaddr**, **cap\_dns\_type\_limit**, **cap\_dns\_family\_limit** - library for getting network host entry in capability mode

**LIBRARY**

library "libcap\_dns"

**SYNOPSIS**

```
#include <sys/nv.h>
#include <libcasper.h>
#include <casper/cap_dns.h>
```

*int*

```
cap_getaddrinfo(cap_channel_t *chan, const char *hostname, const char *servname,
                const struct addrinfo *hints, struct addrinfo **res);
```

*int*

```
cap_getnameinfo(cap_channel_t *chan, const struct sockaddr *sa, socklen_t salen, char *host,
                size_t hostlen, char *serv, size_t servlen, int flags);
```

*struct hostent \**

```
cap_gethostbyname(const cap_channel_t *chan, const char *name);
```

*struct hostent \**

```
cap_gethostbyname2(const cap_channel_t *chan, const char *name, int af);
```

*struct hostent \**

```
cap_gethostbyaddr(const cap_channel_t *chan, const void *addr, socklen_t len, int af);
```

*int*

```
cap_dns_type_limit(cap_channel_t *chan, const char * const *types, size_t ntypes);
```

*int*

```
cap_dns_family_limit(const cap_channel_t *chan, const int *families, size_t nfamilies);
```

**DESCRIPTION**

This service is obsolete and **cap\_net(3)** should be used instead. The **cap\_getaddrinfo()**, and **cap\_getnameinfo()**, functions are preferred over the **cap\_gethostbyname()**, **cap\_gethostbyname2()**, and **cap\_gethostbyaddr()** functions.

The functions `cap_gethostbyname()`, `cap_gethostbyname2()`, `cep_gethostbyaddr()` and `cap_getnameinfo()` are respectively equivalent to `gethostbyname(3)`, `gethostbyname2(3)`, `gethostbyaddr(3)` and `getnameinfo(3)` except that the connection to the `system.dns` service needs to be provided.

The `cap_dns_type_limit()` function limits the functions allowed in the service. The `types` variable can be set to `ADDR2NAME` or `NAME2ADDR`. See the *LIMITS* section for more details. The `nptypes` variable contains the number of *types* provided.

The `cap_dns_family_limit()` functions allows to limit address families. For details see *LIMITS*. The `nfamilies` variable contains the number of *families* provided.

## LIMITS

The preferred way of setting limits is to use the `cap_dns_type_limit()` and `cap_dns_family_limit()` functions, but the limits of service can be set also using `cap_limit_set(3)`. The `nvlist(9)` for that function can contain the following values and types:

type (NV\_TYPE\_STRING)

The *type* can have two values: `ADDR2NAME` or `NAME2ADDR`. The `ADDR2NAME` means that reverse DNS lookups are allowed with `cap_getnameinfo()` and `cap_gethostbyaddr()` functions. In case when *type* is set to `NAME2ADDR` the name resolution is allowed with `cap_getaddrinfo()`, `cap_gethostbyname()`, and `cap_gethostbyname2()` functions.

family (NV\_TYPE\_NUMBER)

The *family* limits service to one of the address families (e.g. `AF_INET`, `AF_INET6`, etc.).

## EXAMPLES

The following example first opens a capability to casper and then uses this capability to create the `system.dns` casper service and uses it to resolve an IP address.

```
cap_channel_t *capcas, *capdns;
int familylimit, error;
const char *ipstr = "127.0.0.1";
const char *typelimit = "ADDR2NAME";
char hname[NI_MAXHOST];
struct addrinfo hints, *res;

/* Open capability to Casper. */
capcas = cap_init();
if (capcas == NULL)
```

```
    err(1, "Unable to contact Casper");

/* Cache NLA for gai_strerror. */
caph_cache_catpages();

/* Enter capability mode sandbox. */
if (caph_enter() < 0)
    err(1, "Unable to enter capability mode");

/* Use Casper capability to create capability to the system.dns service. */
capdns = cap_service_open(capcas, "system.dns");
if (capdns == NULL)
    err(1, "Unable to open system.dns service");

/* Close Casper capability, we don't need it anymore. */
cap_close(capcas);

/* Limit system.dns to reserve IPv4 addresses */
familylimit = AF_INET;
if (cap_dns_family_limit(capdns, &familylimit, 1) < 0)
    err(1, "Unable to limit access to the system.dns service");

/* Convert IP address in C-string to struct sockaddr. */
memset(&hints, 0, sizeof(hints));
hints.ai_family = familylimit;
hints.ai_flags = AI_NUMERICHOST;
error = cap_getaddrinfo(capdns, ipstr, NULL, &hints, &res);
if (error != 0)
    errx(1, "cap_getaddrinfo(): %s: %s", ipstr, gai_strerror(error));

/* Limit system.dns to reverse DNS lookups. */
if (cap_dns_type_limit(capdns, &typelimit, 1) < 0)
    err(1, "Unable to limit access to the system.dns service");

/* Find hostname for the given IP address. */
error = cap_getnameinfo(capdns, res->ai_addr, res->ai_addrlen, hname, sizeof(hname),
    NULL, 0, 0);
if (error != 0)
    errx(1, "cap_getnameinfo(): %s: %s", ipstr, gai_strerror(error));
```

```
printf("Name associated with %s is %s.\n", ipstr, hname);
```

**SEE ALSO**

cap\_enter(2), caph\_enter(3), err(3), gethostbyaddr(3), gethostbyname(3), gethostbyname2(3), getnameinfo(3), capsicum(4), nv(9)

**HISTORY**

The **cap\_dns** service first appeared in FreeBSD 10.3.

**AUTHORS**

The **cap\_dns** service was implemented by Pawel Jakub Dawidek <[pawel@dawidek.net](mailto:pawel@dawidek.net)> under sponsorship from the FreeBSD Foundation.

This manual page was written by  
Mariusz Zaborski <[oshogbo@FreeBSD.org](mailto:oshogbo@FreeBSD.org)>.