NAME

netfuzz — Network Traffic Fuzzer

SYNOPSIS

device netfuzz

DESCRIPTION

IP Packet modifications takes place in the kernel. A pseudo-device, `/dev/netfuzz`, allows userland processes to control the behavior of the modifications through an `ioctl`(2) interface. There are commands to enable and disable the modifications, load rules and retrieve statistics. The most commonly used functions are covered by `netfuzzctl`(8).

IOCTL INTERFACE

netfuzz supports the following `ioctl`(2) commands, available through <netfuzz/netfuzz.h>:

`NETFUZZSTART`
: Start the IP packet modifications.

`NETFUZZSTOP`
: Stop the IP packet modifications.

`NETFUZZCLRRULES`
: Clear all rules.

`NETFUZZSETDEBUG` *`u_int32_t *level`*
: Set the debug level. See the `syslog`(3) man page for a list of valid debug levels.

`NETFUZZSAPPENDRULE` *`struct netfuzz_rule *rule`*

```
#define MAX_FILTER_LEN  1024
#define NETFUZZ_RULES_MAX 8192

struct netfuzz_rule {
    char iface[IFNAMSIZ];
    uint32_t mask;

    char filter[MAX_FILTER_LEN+1];
    struct bpf_program prog;

    uint32_t probability;

    uint32_t offset_start;
    uint32_t offset_end;
        #define DYNOFF_IP_HEADER    (0xff000000+1)
        #define DYNOFF_IP_PAYLOAD   (0xff000000+2)
        #define DYNOFF_TCP_HEADER   (0xff000000+3)
        #define DYNOFF_UDP_HEADER   (0xff000000+4)
        #define DYNOFF_PAYLOAD      (0xff000000+5)
        #define DYNOFF_PACKET_END   (0xff000000+6)

    uint32_t rule_id;
        #define NETFUZZRULE_BITFLIP        1
        #define NETFUZZRULE_BYTEWRITE      2
        #define NETFUZZRULE_BYTEREPLACE    3
        #define NETFUZZRULE_DROP           4
```

```
        #define NETFUZZRULE_DUP             5
        #define NETFUZZRULE_TCPKILL         6


    union {
        struct bitflip {
            uint32_t min;
            uint32_t max;
        } bitflip;

        struct bytewrite {
            uint32_t min;
            uint32_t max;
            uint8_t val;
        } bytewrite;

        struct bytereplace {
            uint32_t min;
            uint32_t max;
            uint8_t old;
            uint8_t new;
        } bytereplace;
    } rule;
};
```

Append *rule* at the end of the current ruleset.

SEE ALSO

`ioctl`(2), `netfuzz.rules`(5), `netfuzzctl`(8),

HISTORY

This is the second generation of the network fuzzing suite of programs that was first implemented as a kernel patch on OpenBSD

AUTHORS