```c
/* Snippet 1 */
char username[8];
int allow = 0;
printf("Enter your username: ");
gets(username);
if (grant_access(username)) {
    allow = 1;
}
if (allow == 1) {
    privileged_action();
}
```

- Purpose of the code:
- Vulnerability:
- Proposed fix:

```c
/* Snippet 2 */
u_int nresp = packet_get_int();
if (nresp > 0) {
    response = xmalloc(nresp * sizeof(char*));
    for (i = 0; i < nresp; i++)
        response[i] = packet_get_string(NULL);
}
```

- Purpose of the code:
- Vulnerability:
- Proposed fix:

```c
/* Snippet 3 */
char *mail_auth(char *mechanism, authresponse_t resp, int argc, char *argv[])
{
    char tmp[MAILTMPLEN];
    AUTHENTICATOR *auth;

    /* make upper case copy of mechanism name */
    ucase(strcpy(tmp, mechanism));

    for (auth = mailauthenticators; auth; auth = auth->next)
        if (auth->server && !strcmp(auth->name, tmp))
            return (*auth->server) (resp, argc, argv);
    return NIL; /* no authenticator found */
}
```

- Purpose of the code:
- Vulnerability:
- Proposed fix:

```c
/* Snippet 4 */
char npath[MAXPATHLEN];
int i;

for (i = 0; *name != '\0' && i < sizeof(npath) - 1; i++, name++)
{
    npath[i] = *name;
    if (*name == '"')
        npath[++i] == '"';
}
npath[i] = '\0'
```

- Purpose of the code:
- Vulnerability:
- Proposed fix:

```c
/* Snippet 5 */
struct header {
    unsigned int length;
    unsigned int message_type;
};

char *read_packet(int sockfd) {
    int n;
    unsigned int length;
    struct header hdr;
    static char buffer[1024];

    if (full_read(sockfd, (void *)&hdr, sizeof(hdr)) <= 0) {
        error("full_read: %m");
        return NULL;
    }

    length = ntohl(hdr.length);
    /* ntohl() converts a given unsigned integer from network byte order
       to host byte order. */

    if (length > (1024 + sizeof(struct header) - 1)) {
        error("not enough room in buffer\n");
        return NULL;
    }

    if (full_read(sockfd, buffer, length - sizeof(struct header)) <= 0) {
        error("read: %m");
        return NULL;
    }

    buffer[sizeof(buffer) - 1] = '\0';

    return strdup(buffer);
}
```

- Purpose of the code:

- Vulnerability:

- Proposed fix: