

## IrfanView 4.57 Denial Of Service / Code Execution

Authored by [Samandeep Singh](#) | Site [sec-consult.com](#)

Posted Feb 17, 2021

IrfanView version 4.57 with WPG.dll version 2.0.0.0 suffer from access violation and out-of-bounds write vulnerabilities that can lead to denial of service or code execution.

tags | [advisory](#), [denial of service](#), [vulnerability](#), [code execution](#)

advisories | [CVE-2021-27224](#)

SHA-256 | [25da92fa817b5a113c55b9e18072698748b07fb0bb80d1febb128c957f5b2d19](#)

[Download](#) | [Favorite](#) | [View](#)

### Related Files

### Share This

Like

Twitter

LinkedIn

Reddit

Digg

StumbleUpon

[Change Mirror](#)[Download](#)

SEC Consult Vulnerability Lab Security Advisory < 20210217-0 >

-----

title: Multiple Vulnerabilities  
product: IrfanView - WPG.dll plugin  
vulnerable version: IrfanView 4.57/WPG.dll version 2.0.0.0  
fixed version: WPG.dll version 3.1.0.0  
CVE number: CVE-2021-27224  
impact: Medium  
homepage: <https://www.irfanview.com>  
found: 2021-02-03  
by: Samandeep Singh (Office Singapore)  
SEC Consult Vulnerability Lab

An integrated part of SEC Consult, an Atos company  
Europe | Asia | North America

<https://www.sec-consult.com>

-----

Vendor description:  
-----

"IrfanView was the first Windows graphic viewer worldwide with Multiple (animated) GIF support. One of the first graphic viewers worldwide with Multipage TIF support. The first graphic viewer worldwide with Multiple ICO support."

Source: [https://www.irfanview.com/main\\_what\\_is\\_engl.htm](https://www.irfanview.com/main_what_is_engl.htm)

Business recommendation:  
-----

SEC Consult recommends upgrading to the latest available version which patches the security issues.

Vulnerability overview/description:  
-----

IrfanView's WPG file parsing library suffers from multiple vulnerabilities. These vulnerabilities can cause application denial of service as well as arbitrary code execution in the worst case scenario. The vulnerabilities can be exploited by an attacker by making the user open a WPG file using IrfanView.

The following vulnerabilities were discovered:

1. Out of Bound Write causing Denial of Service (CVE-2021-27224)
2. Access violation causing Denial of Service while attempting to read from unallocated/freed memory

Note: The vulnerabilities were discovered by fuzzing the WPG.DLL library.

Proof of concept:  
-----

1. Out of Bound Write causing Denial of Service

Below is an excerpt of the decompiled function where the out-of-bound write occurs:

```
signed int __usercall sub_7C42E788@eax(char al@cal, signed int a2)
{
    signed int result; // eax

    switch ( *( _WORD *) (a2 - 10) )
    {
        case 1:
            *( _DWORD *) (a2 - 20) + *( _DWORD *) (a2 - 28) / 8 + *( _DWORD *) (a2 - 24) * *( _DWORD *) (a2 - 4) =
al;
            *( _DWORD *) (a2 - 28) += 8;
            break;
        case 2:
            *( _BYTE *) ( *( _DWORD *) (a2 - 20) + *( _DWORD *) (a2 - 28) / 4 + *( _DWORD *) (a2 - 24) * *( _DWORD *) (a2 - 4) =
al;
            *( _DWORD *) (a2 - 28) += 4;
            break;
        case 4:
            *( _BYTE *) ( *( _DWORD *) (a2 - 20) + *( _DWORD *) (a2 - 28) / 2 + *( _DWORD *) (a2 - 24) * *( _DWORD *) (a2 - 4) =
al;
            *( _DWORD *) (a2 - 28) += 2;
            break;
        case 8:
            *( _BYTE *) ( *( _DWORD *) (a2 - 20) + *( _DWORD *) (a2 - 28) ) ++ + *( _DWORD *) (a2 - 24) * *( _DWORD *) (a2 - 4) =
al;
            break;
    }
    result = *( _DWORD *) (a2 - 28);
    if ( result >= *( unsigned __int16 *) (a2 - 14) )
    {
        *( _DWORD *) (a2 - 28) = 0;
        result = a2;
        --*( _DWORD *) (result - 4);
    }
    return result;
}
```

The vulnerability is triggered in all the cases in the function above.

Also, following excerpt shows the decompiled function which is the caller of above function:

```
int sub_7C4326C()
int v21; // [esp+30h] [ebp-28h]
v0 = 0;
else if ( v1 - 129 < 0x7F )
{
    v3 = v1 - 128;
    do
    {
        sub_7C42E788( *( _BYTE *) (v21 + v0), (signed int) &savedregs);
        --v3;
    }
    while ( v3 );
    ++v0;
}

Below is the Windbg output when a malicious file is opened by IrfanView, along with the result of windbg.

Exploitable plugin:



```
0:000> g
(snip)
```


```

### File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr
Sa					
				1	2
3					
4	5	6	7	8	9
10					
11	12	13	14	15	16
17					
18	19	20	21	22	23
24					
25	26	27	28	29	30
31					

### Top Authors In Last 30 Days

Red Hat 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11security 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 files
Julien Ahrens 4 files
T. Weber 4 files

### File Tags

ActiveX (932)	December 2022
Advisory (79,754)	November 2022
Arbitrary (15,694)	October 2022
BBS (2,859)	September 2022
Bypass (1,619)	August 2022
CGI (1,018)	July 2022
Code Execution (8,926)	June 2022
Conference (673)	May 2022
Cracker (840)	April 2022
CSRF (3,290)	March 2022
DoS (22,602)	February 2022
Encryption (2,349)	January 2022
Exploit (50,359)	Older
File Inclusion (4,165)	

### File Upload (946)

Firewall (821)	AIX (426)
Info Disclosure (2,660)	Apple (1,926)
Intrusion Detection (867)	BSD (370)
Java (2,899)	CentOS (55)
JavaScript (821)	Cisco (1,917)
Kernel (6,291)	Debian (6,634)
Local (14,201)	Fedora (1,600)
Magazine (586)	FreeBSD (1,242)
Overflow (12,419)	Gentoo (4,272)
Perl (1,418)	HPUX (878)
PHP (5,093)	iOS (330)
Proof of Concept (2,291)	iPhone (108)
Protocol (3,435)	IRIX (220)
Python (1,467)	Juniper (67)
Remote (30,044)	Linux (44,315)
Root (3,504)	Mac OS X (684)
Ruby (594)	Mandriva (3,105)
Scanner (1,631)	NetBSD (255)
Security Tool (7,777)	OpenBSD (479)
Shell (3,103)	RedHat (12,469)
Shellcode (1,204)	Slackware (941)
Sniffer (886)	Solaris (1,607)

```

ModLoad: 08340000 0835d000 D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
ModLoad: 08340000 0835d000 D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
ModLoad: 704b0000 704f2000 C:\WINDOWS\SysWOW64\WINSTA.dll
(32f0.c58): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
*** WARNING: Unable to verify checksum for D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
eax=16640c9f ebx=0000002b ecx=166413e0 edx=ffffec14 esi=000004fb edi=08356c80
eip=08352ec6 esp=00197f4f ebp=00197f44 iopl=0         nv up ei ng zr na pe nc
cs=0023  as=002b  ds=002b  es=002b  fs=0053  gs=002b  efl=00210286
WPG+0x12ec6:
08352ec6 880411 mov byte ptr [ecx+edx],al ds:002b:1663fff4??
0:00D> kv
# ChildEBP RetAddr  Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
00 001973f4 083534b7 00197470 0019740c 083534fc WPG+0x12ec6
01 00197470 083534b4 00197484 08353b0a 001974b4 WPG+0x134b7
02 001974b4 00486731 005a7c00 00197dac 001981bc WPG!ReadWPG_W+0x214
03 00197dc8 77c4b2e3 51162351 75dc4e5e 107e28f0 Image00400000+0x86731
04 00197e70 70edc3ce 00198014 77be2c00 f7e1cf3f ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
05 00197e98 77c4b834 03faa7c8 00000000 70edad40 verifier!AVrpfDphWritePageHeapBlockInformation+0x9e (FPO: [Non-Fpo])
06 00197e88 77c4b2e3 d8c27e53 07604000 07572b60 ntdll!RtlpStdLogCapturedStackTrace+0xfa (FPO: [Non-Fpo])
07 00197f5c 00197f6c 70ed7f5a 075716cc 00197f98 ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
08 00197f6c 70ed9822 70ed9848 70ed7f5a 075716cc 0x197f6c
09 00197f9c 70edae2f 07571000 07572b60 07571000 verifier!AVrpfDphPlaceOnDelayFree+0x262 (FPO: [Non-Fpo])
0a 00197fb4 77c52ca1 07570000 77bee5ba 77c52f11 verifier!AVrpfDebugPageHeapFree+0xef (FPO: [Non-Fpo])
0b 00198024 77bb3c45 07604000 803eb45f 00000000 ntdll!RtlDebugFreeHeap+0x3e (FPO: [Non-Fpo])
0c 001981cc 00450020 0063006e 0064006f 00660069 ntdll!RtlpFreeHeap+0xd5 (FPO: [Non-Fpo])
0d 001981f4 004d51a9 07577bf8 00000000 00596620 Image00400000+0x50020
0e 00198ab0 00000000 00000000 00000000 00000001 Image00400000+0xd51a9
0:00D> !msec.exploitable

!exploitable 1.6.0.0
Exploitability Classification: EXPLOITABLE
Recommended Bug Title: Exploitable - User Mode Write AV starting at WPG+0x00000000000012ec6
(Hash=0x7d95926e.0x254455d2)

User mode write access violations that are not near NULL are exploitable.

2. Access violation causing Denial of Service while attempting to read from
unallocated/freed memory

Example 1:
-----
Below is an excerpt of the decompiled function where the access violation occurs:

DWORD ReadWPG_W(int a1&ebx), int a2&edi), int a3&esi), int a4, wchar_t *a5, wchar_t *a6)
{
    [SNIP]
    v9 = (*(int (__fastcall *) (System::TObject *, void *, signed int)))(*(__DWORD *)dword_7C4687C + 12)) (
        dword_7C4687C,
        sunk_7C46C8D,
        1);
    dword_7C46C84 = sub_7C42AB8(v9);

    [SNIP]
}

In the above code, the address of "dword_7C4687C + 12()" is pointing to a memory
location which is freed or unallocated and the exception occurs.

Below is the Windbg output when a malicious file is opened by IrfanView, along with
the result of windbg.

Exploitable plugin:

0:00D> g
[SNIP]
ModLoad: 07420000 0743d000 D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
ModLoad: 07420000 0743d000 D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
ModLoad: 704b0000 704f2000 C:\WINDOWS\SysWOW64\WINSTA.dll
(1d38.313e): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
*** WARNING: Unable to verify checksum for D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
eax=16420b10 ebx=07436c80 ecx=00000001 edx=07436c80 esi=f0f0f0f0 edi=07436c80
eip=07433933 esp=00197478 ebp=001974b4 iopl=0         nv up ei pl zr na pe nc
cs=0023  as=002b  ds=002b  es=002b  fs=0053  gs=002b  efl=00210246
WPG!ReadWPG_W+0x1333:
07433933 ff560c call dword ptr [esi+0Ch] ds:002b:f0f0f0fc=????????
0:00D> kv
# ChildEBP RetAddr  Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
00 001974b4 00486731 005a7c00 00197dac 001981bc WPG!ReadWPG_W+0x133
01 00197dc8 77c4b2e3 51162351 75dc4e5e 100f2374 Image00400000+0x86731
02 00197e70 70edc3ce 00198014 77be2c00 90be46da ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
03 00197e98 77c4b834 03fca7c8 00000000 70edad40 verifier!AVrpfDphWritePageHeapBlockInformation+0x9e (FPO: [Non-Fpo])
04 00197e88 77c4b2e3 d8c27e53 07624000 07592b60 ntdll!RtlpStdLogCapturedStackTrace+0xfa (FPO: [Non-Fpo])
05 00197f5c 00197f6c 70ed7f5a 075916cc 00197f98 ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
06 00197f6c 70ed9822 70ed9848 70ed7f5a 0x197f6c
07 00197f9c 70edae2f 07591000 07592b60 07591000 verifier!AVrpfDphPlaceOnDelayFree+0x262 (FPO: [Non-Fpo])
08 00197fb4 77c52ca1 07590000 77bee5ba 77c52f11 verifier!AVrpfDebugPageHeapFree+0xef (FPO: [Non-Fpo])
09 00198024 77bb3c45 07624000 e7613dba 00000000 ntdll!RtlDebugFreeHeap+0x3e (FPO: [Non-Fpo])
0a 001981cc 00450020 0063006e 0064006f 00660069 ntdll!RtlpFreeHeap+0xd5 (FPO: [Non-Fpo])
0b 001981f4 004d51a9 07597bf8 00000000 00596620 Image00400000+0x50020
0c 00198ab0 00000000 00000000 00000000 00000001 Image00400000+0xd51a9
0:00D> !msec.exploitable

!exploitable 1.6.0.0
Exploitability Classification: EXPLOITABLE
Recommended Bug Title: Exploitable - Read Access Violation on Control Flow starting at
WPG!ReadWPG_W+0x0000000000000133 (Hash=0x57561ac2.0x7ef88dfa)

Access violations not near null in control flow instructions are considered exploitable.

Example 2:
-----
Below is an excerpt of the decompiled function where the access violation occurs:

signed int __fastcall System::SysFreeMem(void *a1){
    [SNIP]

    v10 = (__DWORD *)((char *)v11 + v4); // exception occurs here.
    if ( (__DWORD *)((char *)v11 + v4) != (__DWORD *)dword_7C46618 )
    if (*v10 & 2 )
    {
        if ( (*v10 & 0x7FFFFFFF) < 4 )
        {
            dword_7C465C0 = 11;
            goto LABEL_29;
        }
        *v10 |= 1u;
    }
    [SNIP]
}

Also, following excerpt shows the decompiled function which is the caller of above
function:

int __fastcall System::linkproc__ FreeMem(int a1)
{
    int v1; // eax
    int v2; // ebx

    if ( !a1 )
    return 0;
    v1 = off_7C4503C(); //System::SysFreeMem(void *a1) - calling the above function here.
    v2 = v1;
    if ( v1 )
    {
        LOBYTE(v1) = 2;
        System::Error(v1);
    }
    return v2;
}

Below is the Windbg output when a malicious file is opened by IrfanView, along with
the result of windbg.

Exploitable plugin:

0:00D> g
[SNIP]
ModLoad: 083f0000 0840d000 D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
ModLoad: 704b0000 704f2000 C:\WINDOWS\SysWOW64\WINSTA.dll
(2740.70c): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.

```

Spoof (2,166)	SUSE (1,444)
SQL Injection (16,102)	Ubuntu (8,199)
TCP (2,379)	UNIX (9,159)
Trojan (686)	UnixWare (185)
UDP (676)	Windows (6,511)
Virus (662)	Other
Vulnerability (31,136)	
Web (9,365)	
Whitepaper (3,729)	
x86 (946)	
XSS (17,494)	
Other	

```
*** WARNING: Unable to verify checksum for D:\Softwares\IrfanView_downloads\IView457_32\Plugins\WPG.DLL
eax=166c0c38 ebx=0001000c ecx=166b0c2c edx=00000008 esi=00000007 edi=08406c80
eip=083f2634 esp=001973d8 ebp=001973f8 iopl=0         nv up ei pl zr ac pe nc
cs=0023  as=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00210216
WPG+0x2634:
083f2634 8b00 mov eax,dword ptr [eax] ds:002b:166c0c38=????????
0:000> kv
# ChildEBP RetAddr  Args to Child
WARNING: Stack unwinding information not available. Following frames may be wrong.
00 001973f8 083f29a3 0019742c 083f3f9e 00000775 WPG+0x2634
01 00197470 08403a14 00197484 08403b1a 001974b4 WPG+0x29a3
02 001974b4 00486731 005a7c00 00197dac 001981bc WPG!ReadWPG_M+0x214
03 001974b8 77c4b2a3 51162351 75d4e4fe 17872f3c Image00400000+0x86731
04 00197470 70edc3ce 00198014 77be2c00 67f49c30 ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
05 00197e98 77c4b834 0403a7c8 00000000 70edad40 verifier!AVrpfDphWritePageHeapBlockInformation+0x9e (FPO:
[Non-Fpo])
06 00197ba8 77c4b2a3 d8c27a53 07764000 076d2b60 ntdll!RtlpStdLogCapturedStackTrace+0xfa (FPO: [Non-Fpo])
07 00197f5c 00197f6c 70ed7f5a 076d16cc 00197f98 ntdll!RtlStdLogStackTrace+0x43 (FPO: [Non-Fpo])
08 00197f6c 70ed9822 70ed9848 70ed7f5a 076d16cc 0x197f6c
09 00197f9c 70edae2f 076d1000 076d2b60 076d1000 verifier!AVrpfDphPlaceOnDelayFree+0x262 (FPO: [Non-Fpo])
0a 00197fb4 77c520a1 076d0000 77bee5ba 77c52f1f verifier!AVrpfDebugPageHeapFree+0xef (FPO: [Non-Fpo])
0b 00198024 77bb3c45 07764000 102be750 00000000 ntdll!RtlpDebugFreeHeap+0x3e (FPO: [Non-Fpo])
0c 001981cc 00450020 0063006e 0064006f 006e0069 ntdll!RtlpFreeHeap+0xd5 (FPO: [Non-Fpo])
0d 001981f4 004d51a9 076d7bf8 00000000 00596620 Image00400000+0x50020
0e 00198ab0 00000000 00000000 00000000 00000001 Image00400000+0xd51a9
0:000> !msec.exploitable

!exploitable 1.6.0.0
Exploitability Classification: UNKNOWN
Recommended Bug Title: Data from Faulting Address controls Branch Selection starting at
WPG+0x00000000000002634 (Hash=0x7d95926e.0xbc07e85c)

The data from the faulting address is later used to determine whether or not a branch
is taken.

Vulnerable / tested versions:
-----
The following version has been tested which was the latest version available at the
time of the test.

* IrfanView 4.57/WPG.dll version 2.0.0.0 (Both x86 & x64 versions)

Vendor contact timeline:
-----
2021-02-07 | Contacting vendor with details of vulnerabilities through irfanview@gmx.net
2021-02-08 | Vendor acknowledged the email and mentioned that fixed plugin will be available soon
2021-02-12 | Vendor shared the new plugin with fixes
2021-02-17 | Coordinated release of security advisory

Solution:
-----
It's recommended to update the WPG plugin to it's latest version 3.1.0.0:
https://www.irfanview.com/plugins.htm

Direct link for IrfanView 32 bit WPG plugin:
https://www.irfanview.net/plugins/wpg_32.zip

Direct link for IrfanView 64 bit WPG plugin:
https://www.irfanview.net/plugins/wpg_64.zip

Workaround:
-----
None

Advisory URL:
-----
https://sec-consult.com/vulnerability-lab/

-----
SEC Consult Vulnerability Lab

SEC Consult, an Atos company
Europe | Asia | North America

About SEC Consult Vulnerability Lab
The SEC Consult Vulnerability Lab is an integrated part of SEC Consult, an
Atos company. It ensures the continued knowledge gain of SEC Consult in the
field of network and application security to stay ahead of the attacker. The
SEC Consult Vulnerability Lab supports high-quality penetration testing and
the evaluation of new offensive and defensive technologies for our customers.
Hence our customers obtain the most current information about vulnerabilities
and valid recommendation about the risk profile of new technologies.

-----
Interested to work with the experts of SEC Consult?
Send us your application https://sec-consult.com/career/

Interested in improving your cyber security with the experts of SEC Consult?
Contact our local offices https://sec-consult.com/contact/

-----
Mail: research at sec-consult dot com
Web: https://www.sec-consult.com
Blog: http://blog.sec-consult.com
Twitter: https://twitter.com/sec_consult

EOF Samandeep Singh / #2021
```

[Login](#) or [Register](#) to add favorites

**packet storm**

© 2022 Packet Storm. All rights reserved.

### Site Links


News by Month
News Tags
Files by Month
File Tags
File Directory


### About Us

History & Purpose
Contact Information
Terms of Service
Privacy Statement
Copyright Information

### Hosting By

Rokasec
---------

 Follow us on Twitter

 Subscribe to an RSS Feed