ESPILON
SECURITY RESEARCH

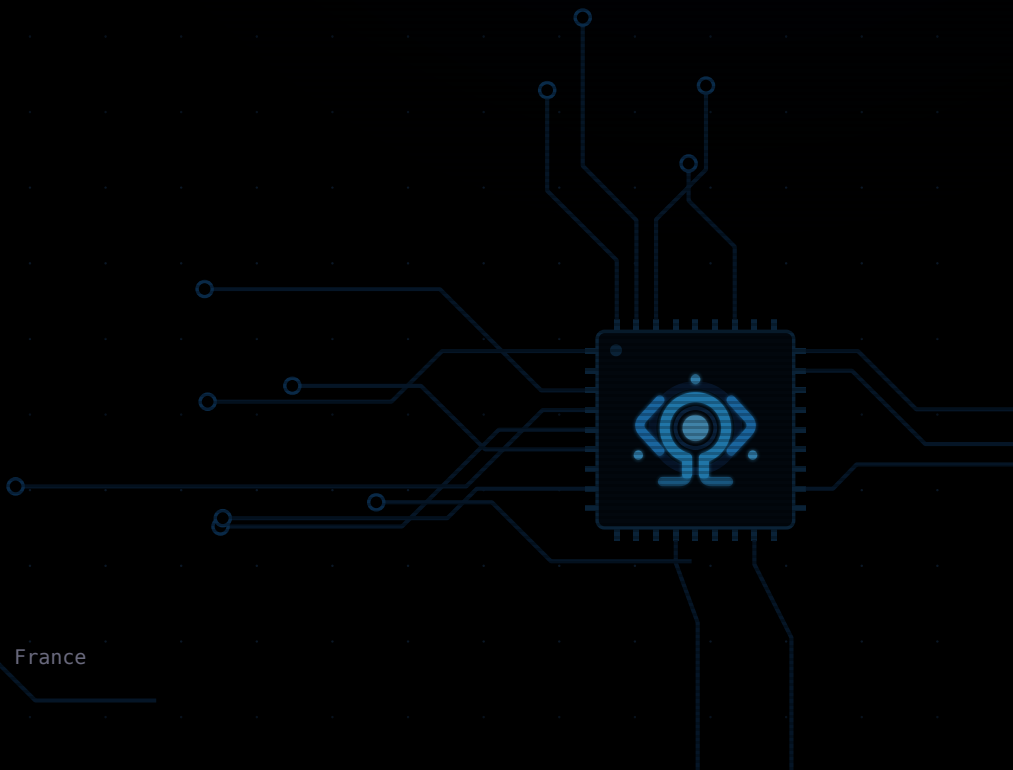
SECURITY ADVISORY

Out-of-bounds write in the X.Org server EDID DS_RANGES handler

CVE-2026-109091 NOT SCORED

X.Org X server · xf86 modes (EDID)

CVE PUBLISHED **10 Nov 2026**



Marouane El Moufid (Eun0us)

espilon.net · Association loi 1901 · France

01 Vulnerability

N/A

NOT SCORED

CVSS · X.ORG ADVISORY

No CVSS score published with the advisory

The X.Org advisory of 2026-11-10 carries no score, and the CVE record was not yet published on cve.org when this report was written (checked 2026-11-10). Our own assessment is given in section 02.

CVE	CVE-2026-109091, X.Org Security Advisory, 2026-11-10
Component	`handle_detailed_monset()` in `hw/xfree86/modes/xf86EdidModes.c`, DS_RANGES branch
Type	Out-of-bounds write (CWE-787) past the fixed-size `hsync[]` / `vrefresh[]` arrays of `MonRec`
Attacker	Physical: a monitor, or a device in the display path (KVM, dock, EDID emulator), that presents a crafted EDID
Fix	`95cf222d` (xorg/xserver), released in xorg-server 21.1.26
Credit	Eun0us / Espilon

Summary

When the X server reads a monitor's EDID, `handle_detailed_monset()` records each Display-Range descriptor (`DS_RANGES`) into the monitor record. The horizontal-sync and vertical-refresh ranges are stored at index `nHsync` / `nVrefresh`, and the counter is incremented, with no check against the array capacity (`MAX_HSYNC` and `MAX_VREFRESH`, both 8). The ranges coming from `xorg.conf` are bounded; the ones coming from the monitor are not. An EDID that carries more than eight such descriptors, across the base block and CEA extension blocks, writes past the arrays and over the members that follow them in `MonRec`, among them the `Modes` and `Last` mode-list pointers and the `DDC` pointer.

Affected and fixed versions

Per the X.Org advisory: X.Org X server before xorg-server 21.1.26. The advisory header covers the whole batch, Xwayland included; this code lives under `hw/xfree86`, which Xwayland does not use, so as far as we can tell Xwayland is not affected by this issue.

AFFECTED	FIXED
`xorg-server < 21.1.26`	`xorg-server 21.1.26, commit 95cf222d`

Timeline

DATE	EVENT
2026-07-17	Report sent to the X.Org security team, with a host ASan harness
2026-10-02	Fix authored by Olivier Fourdan (`95cf222d`)
2026-10-06	Report acknowledged by X.Org
2026-10-27	Draft advisory and fixes shared with the reporters for review
2026-11-03	Fix and advisory text reviewed by Espilon
2026-11-10	Advisory published, xorg-server 21.1.26 released

02 Technical analysis

Vulnerable code

hw/xfree86/modes/xf86EdidModes.c at b1678490 (parent of 95cf222d)

```
1119  handle_detailed_monset(struct detailed_monitor_section *det_mon, void *data)
1120  {
1121      int clock;
1122      struct det_mon_parameter *p = (struct det_mon_parameter *) data;
1123      int scrnIndex = ((xf86MonPtr) (p->Monitor->DDC))->scrnIndex;
1124
1125      switch (det_mon->type) {
1126      case DS_RANGES:
1127          if (!p->have_hsync) {
1128              if (!p->Monitor->nHsync)
1129                  xf86DrvMsg(scrnIndex, X_INFO,
1130                             "Using EDID range info for horizontal sync\n");
1131              p->Monitor->hsync[p->Monitor->nHsync].lo =
1132                  det_mon->section.ranges.min_h;
1133              p->Monitor->hsync[p->Monitor->nHsync].hi =
1134                  det_mon->section.ranges.max_h;
1135              p->Monitor->nHsync++;
1136          }
...
1142          if (!p->have_vrefresh) {
1143              if (!p->Monitor->nVrefresh)
1144                  xf86DrvMsg(scrnIndex, X_INFO,
1145                             "Using EDID range info for vertical refresh\n");
1146              p->Monitor->vrefresh[p->Monitor->nVrefresh].lo =
1147                  det_mon->section.ranges.min_v;
1148              p->Monitor->vrefresh[p->Monitor->nVrefresh].hi =
1149                  det_mon->section.ranges.max_v;
1150              p->Monitor->nVrefresh++;
1151          }
```

The destination arrays have eight slots each, and the pointers the overflow reaches sit right after them:

hw/xfree86/common/xf86str.h at b1678490 (abridged)

```
105   #define MAX_HSYNC 8
106   #define MAX_VREFRESH 8
...
129       int nHsync;
130       range hsync[MAX_HSYNC];
131       int nVrefresh;
132       range vrefresh[MAX_VREFRESH];
133       DisplayModePtr Modes;       /* Start of the monitor's mode list */
134       DisplayModePtr Last;        /* End of the monitor's mode list */
...
139       void *DDC;
```

Root cause

`xf86EdidMonitorSet()` sets `have_hsync` and `have_vrefresh` only when ranges were already configured (`nHsync != 0`, lines 1195-1196), then walks every detailed descriptor of the EDID with `xf86ForEachDetailedBlock()` (line 1198). The walk visits the four descriptors of the base block, then the detailed descriptors of each CEA-861 extension block:

hw/xfree86/ddc/interpret_edid.c at b1678490, xf86ForEachDetailedBlock()

```
246       for (i = 0; i < DET_TIMINGS; i++)
247           fn(mon->det_mon + i, data);
248
249       for (i = 0; i < mon->no_sections; i++) {
250           ext = mon->rawData + EDID1_LEN * (i + 1);
251           switch (ext[EXT_TAG]) {
252           case CEA_EXT:
253               handle_cea_detail_block(ext, mon, fn, data);
254               break;
```

Every descriptor of type `DS_RANGES` adds one entry to each array. The number of descriptors is chosen by the EDID, so a monitor with enough CEA extension blocks pushes both counters past 8. From the ninth descriptor on, the stores land beyond the arrays: past `hsync[]` into `nVrefresh` and `vrefresh[]`, past `vrefresh[]` into `Modes`, `Last` and the members that follow. The xorg.conf path, which fills the same arrays, already stops at the capacity.

Reachability

The X server parses EDID on its own when it detects an output, over DDC or from the kernel. No user action is needed beyond connecting the display. The attacker needs control of the EDID the server reads: a monitor, or a device inserted

in the display path such as a KVM switch, a dock or an EDID emulator. The overflow requires that `xorg.conf` does not set `HorizSync` / `VertRefresh`, which is the default.

Impact

The values written are EDID-derived sync and refresh limits, stored as floats. They land on the `Modes`, `Last` and `DDC` pointers, which the server uses afterwards. The expected result is a crash of the X server, a privileged process on many systems, from a malicious display. The attacker controls float-shaped values (for example `0x42fc0000`, which is `126.0f`), not arbitrary 64-bit pointers. We demonstrated the write and the pointer overwrite in a host harness; we did not test a physical monitor and we make no code-execution claim.

Our assessment at report time: CVSS 3.1 `AV:P/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:H`, base score 6.1 (medium). This is not the score of a CNA.

Proof

A host harness compiles the real `interpret_edid.c` and the `handle_detailed_monset()` function verbatim from xorg/xserver at `588a546e` (SHA-1 of each file recorded with the harness), over a `MonRec` with the server's layout (224 bytes). Only `xf86DrvMsg()` and the allocator are stubbed. A benign EDID with 4 DS_RANGES descriptors walks cleanly. A crafted EDID with 64 descriptors across 10 CEA extensions faults on a write inside `handle_detailed_monset()`. Output, trimmed (libc frames removed):

• AddressSanitizer: WRITE in handle_detailed_monset (64 DS_RANGES, 10 extensions), trimmed

```
[harness] mode=wild n_ext=10 DS_RANGES=64 edid_len=1408 sizeof(MonRec)=224 MAX_HSYNC=8
==3112461==ERROR: AddressSanitizer: SEGV on unknown address 0x611215a000a4
==3112461==The signal is caused by a WRITE memory access.
    #0 in handle_detailed_monset handle_detailed_monset.inc:37:60
    #1 in handle_cea_detail_block interpret_edid.c:234:9
    #2 in xf86ForEachDetailedBlock interpret_edid.c:253:13
```

With 22 descriptors over 3 extensions, the walk completes and the harness prints the monitor record afterwards: `Modes` and `Last` both hold `0x42fc0000`, a value taken from the EDID. The same 64-descriptor input built without a sanitizer exits with `SIGSEGV` (exit code 139).

03 Fix and recommendation

Upstream fix

Commit `95cf222d` (Olivier Fourdan, merge request 2389) bounds each store by the array capacity and drops the extra descriptors, while still applying the maximum pixel clock. Hsync hunk, verbatim (the vrefresh hunk is identical with `MAX_VREFRESH`):

• xorg/xserver commit 95cf222d43c9, hw/xfree86/modes/xf86EdidModes.c (first hunk)

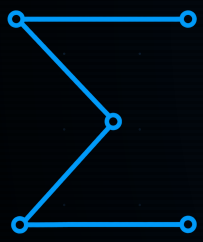
```
@@ -1125,14 +1125,16 @@ handle_detailed_monset(struct detailed_monitor_section *det_mon, void *data)
     switch (det_mon->type) {
     case DS_RANGES:
         if (!p->have_hsync) {
-            if (!p->Monitor->nHsync)
-                xf86DrvMsg(scrnIndex, X_INFO,
-                           "Using EDID range info for horizontal sync\n");
-            p->Monitor->hsync[p->Monitor->nHsync].lo =
-                det_mon->section.ranges.min_h;
-            p->Monitor->hsync[p->Monitor->nHsync].hi =
-                det_mon->section.ranges.max_h;
-            p->Monitor->nHsync++;
+            if (p->Monitor->nHsync < MAX_HSYNC) {
+                if (!p->Monitor->nHsync)
+                    xf86DrvMsg(scrnIndex, X_INFO,
+                               "Using EDID range info for horizontal sync\n");
+                p->Monitor->hsync[p->Monitor->nHsync].lo =
+                    det_mon->section.ranges.min_h;
+                p->Monitor->hsync[p->Monitor->nHsync].hi =
+                    det_mon->section.ranges.max_h;
+                p->Monitor->nHsync++;
+            }
         }
```

Recommendation

Update to xorg-server 21.1.26 or later, or to your distribution's package carrying commit `95cf222d`. Until then, setting `HorizSync` and `VertRefresh` in the `Monitor` section of `xorg.conf` makes the server ignore the EDID ranges, and the overflowing branch is not taken. Do not connect untrusted displays or display adapters to machines running an unpatched Xorg session.

04 Credit and references

- Credit: Eun0us / Espilon (Marouane El Moufid), as listed in the X.Org advisory and the fix commit.
- X.Org Security Advisory, 2026-11-10: lists.x.org/archives/xorg-announce/2026-November/003759.html
- Fix: gitlab.freedesktop.org/xorg/xserver/-/commit/95cf222d
- Release: xorg-server 21.1.26, lists.x.org/archives/xorg-announce/2026-November/003760.html
- CVE record: www.cve.org/CVERecord?id=CVE-2026-109091



ESPILON

SECURITY RESEARCH

Embedded, IoT and OT security research

espilon.net

eun0us@espilon.net

Association loi 1901 · France