FIELD EFFECT

Tales From The Crypt

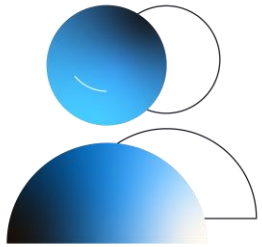
Bug Hunting in the Windows CryptoAPI

RECon 2024

Background

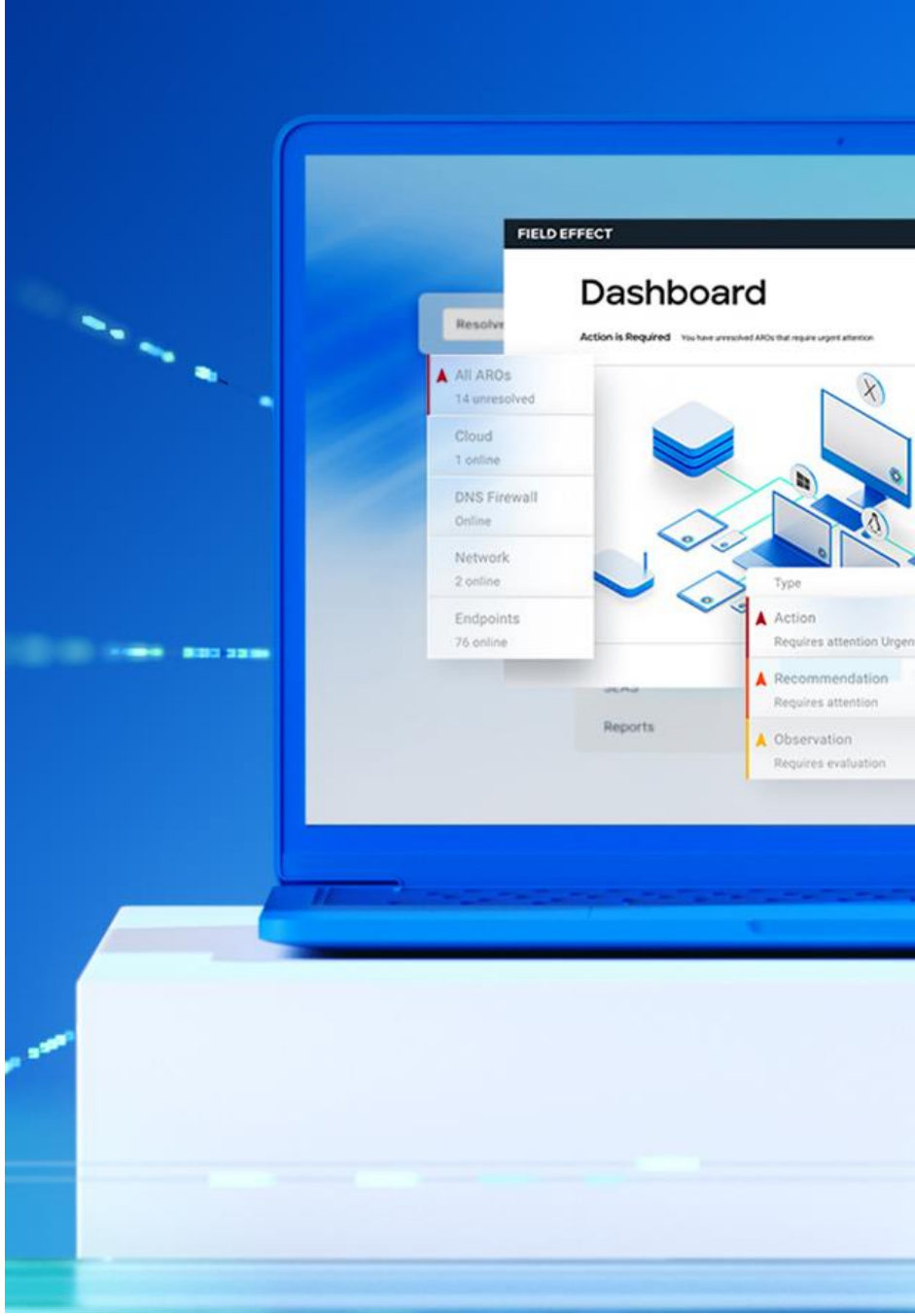
Field Effect

- Founded in 2016, ~200 employees.
- Headquartered in Ottawa, Canada with employees also in AU, NZ, UK and US.
- A holistic, comprehensive approach to cyber security.
- A focus on solving security challenges for small and mid-size organizations.



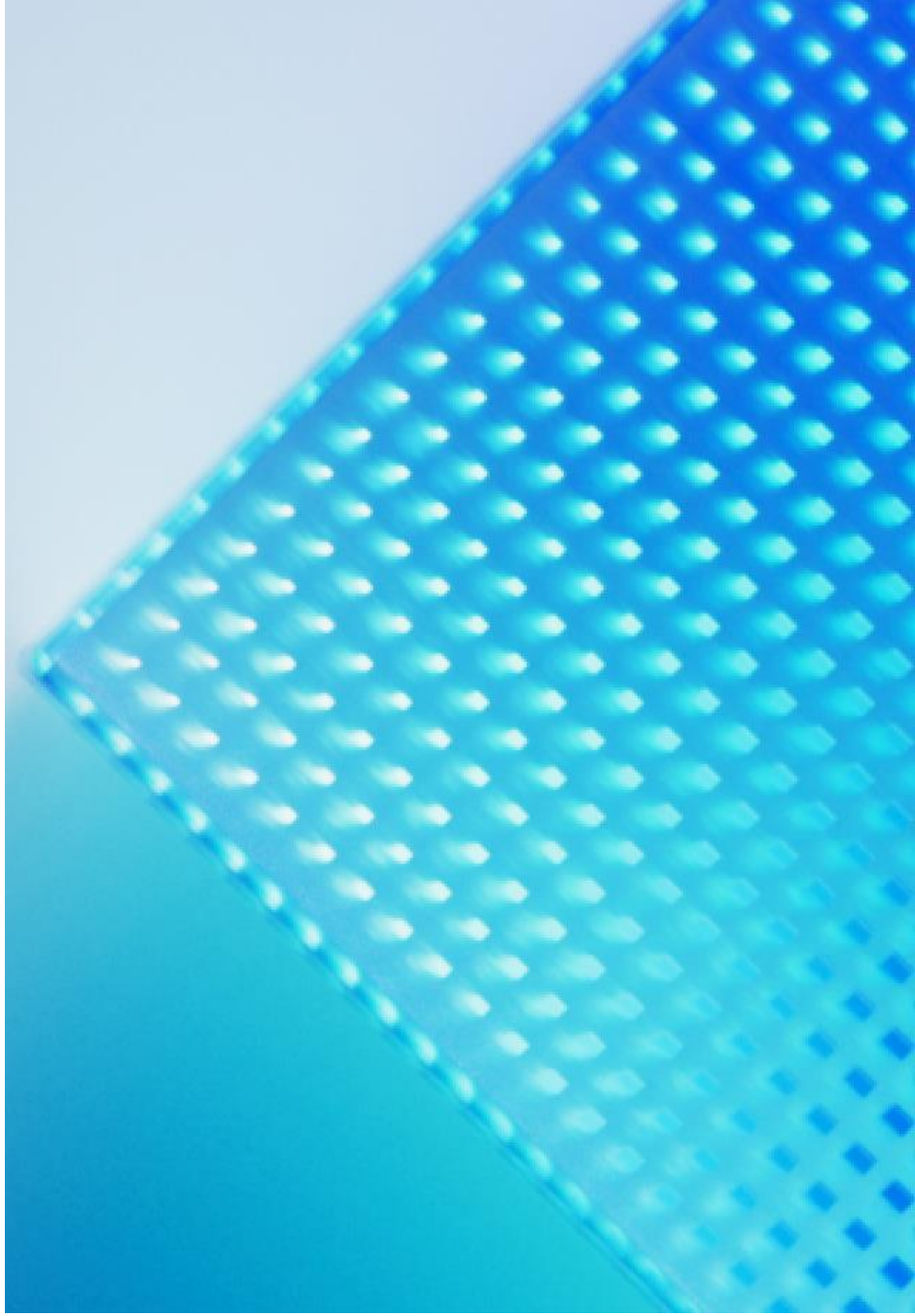
Erik Egsgard

- ~20 years in cyber security
- EDR Security Developer
- Vulnerability researcher



Story Time

```
- Crypt Decode fuzzing:

  - Got an AFL crash somewhere, think is is in the decoding of
 szOID_PKIX_POLICY_QUALIFIER_USERNOTICE

  - Another one in decoding of CRYPT_TIMESTAMP_RESPONSE

                                                4,31          Top
```



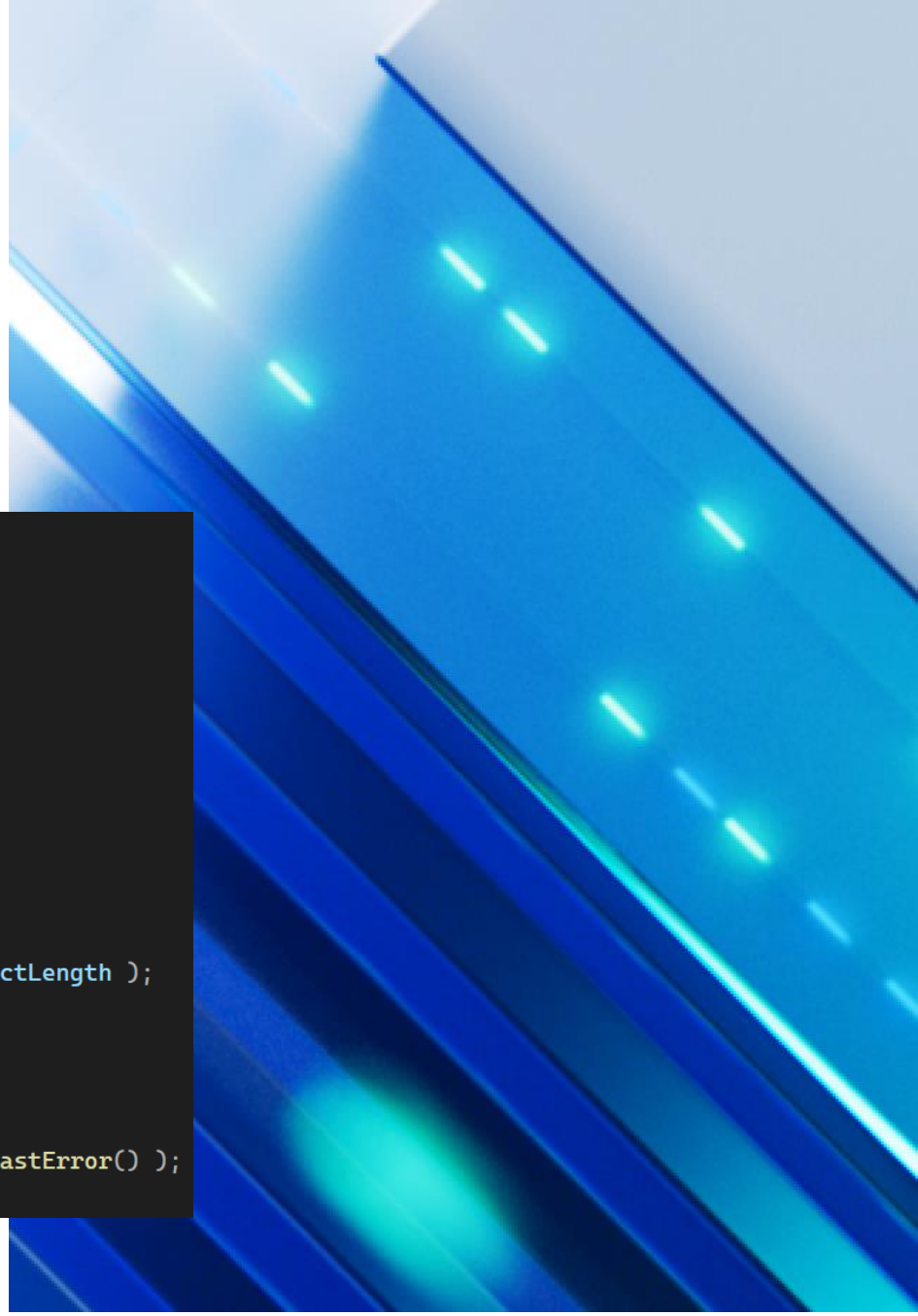
FIELD EFFECT

Crypt Decoding

C++

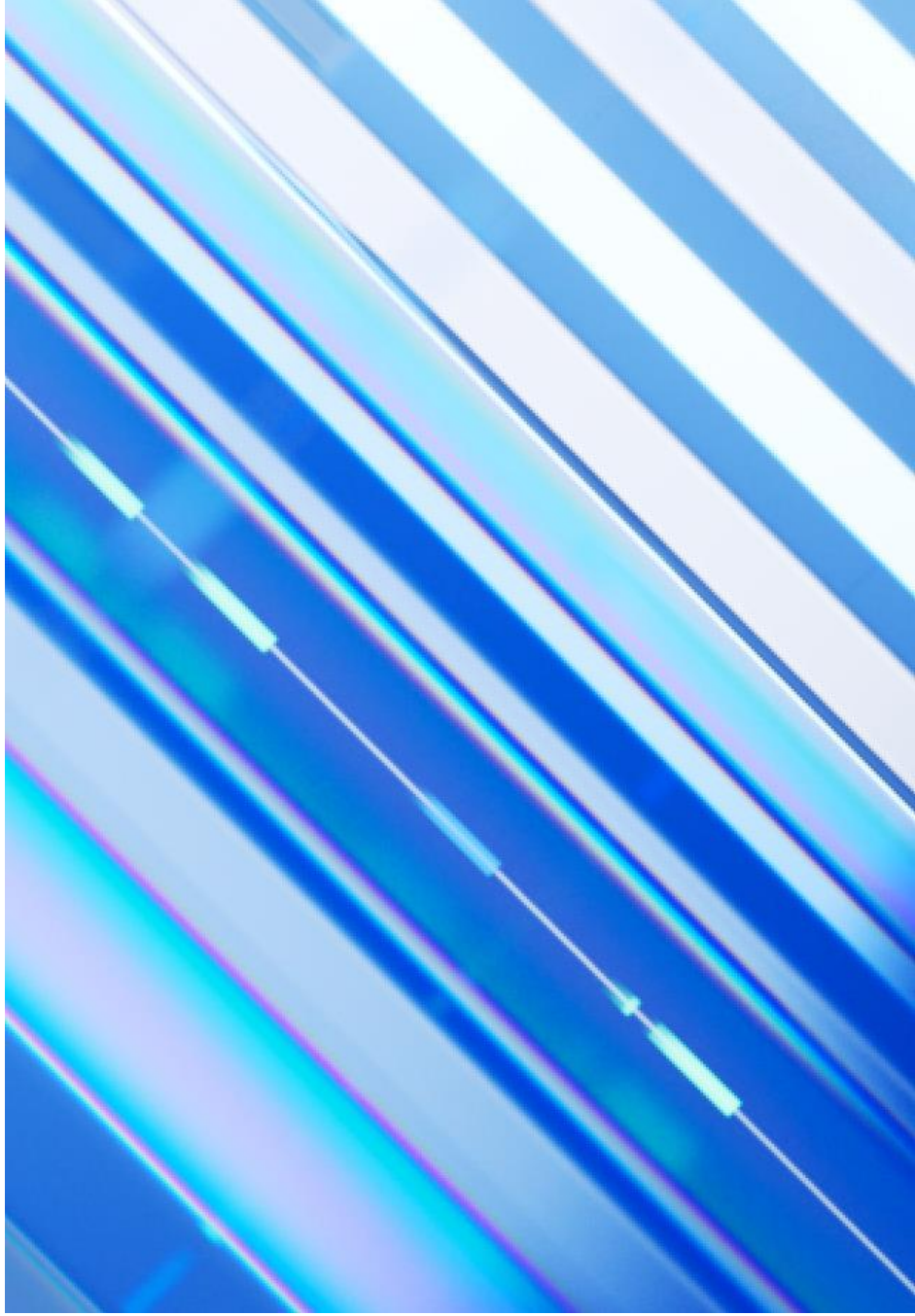
```
BOOL CryptDecodeObjectEx(
  [in]      DWORD              dwCertEncodingType,
  [in]      LPCSTR             lpszStructType,
  [in]      const BYTE         *pbEncoded,
  [in]      DWORD              cbEncoded,
  [in]      DWORD              dwFlags,
  [in]      PCRYPT_DECODE_PARA pDecodePara,
  [out]     void               *pvStructInfo,
  [in, out] DWORD              *pcbStructInfo
);
```

```
decodeParam.cbSize = sizeof( decodeParam );
decodeParam.pfnAlloc = allocWrapper;
decodeParam.pfnFree = freeWrapper;

if( CryptDecodeObjectEx(
    X509_ASN_ENCODING | PKCS_7_ASN_ENCODING,
    targetObjectType,
    decodePtr,
    dataLength,
    CRYPT_DECODE_ALLOC_FLAG,
    &decodeParam,
    &object,
    &objectLength ) )
{
    vlog( "Decoded object into 0x%08x bytes\n", objectLength );

    freeWrapper( object );
}
else
{
    vlog( "CryptDecodeObjectEx failed: %08x\n", GetLastError() );
}
```



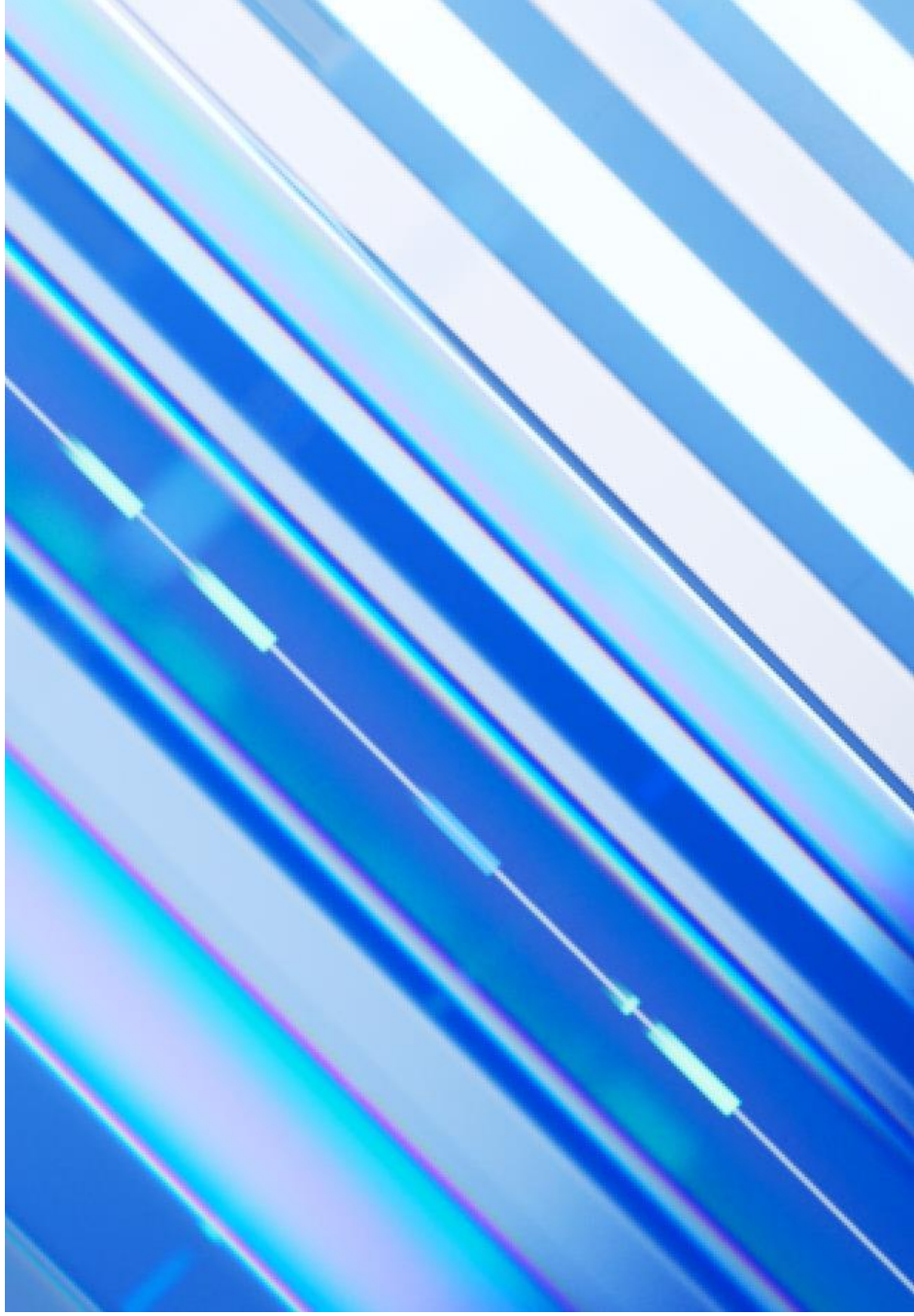
AFL False Start

```
WinAFL 1.17 by <ifratric@google.com>
Based on AFL 2.43b by <lcamtuf@google.com>
[+] You have 4 CPU cores with average utilization of 0%.
[+] Try parallel jobs - see afl_docs\parallel_fuzzing.txt.
[*] Checking CPU core loadout...
[+] Found a free CPU core, binding to #0.
[+] Process affinity is set to 1.
[*] Setting up output directories...
[+] Output directory exists but deemed OK to reuse.
[*] Deleting old session data...
[+] Output dir cleanup successful.
[*] Scanning 'in_x509'...
[+] No auto-generated dictionary tokens to reuse.
[*] Creating hard links for all input files...
[*] Attempting dry run with 'id_000000'...
Instrumented module crypt32.dll, code size: 1101824
Instrumented module msasn1.dll, code size: 45056
*** Heap Corruption - CRASHING ***
Exception at address 00007FF72AB21476
Access address: 0000000000000000

[-] Oops, the program crashed with one of the test cases provided. There are
    several possible explanations:

    - The test case causes known crashes under normal working conditions. If
      so, please remove it. The fuzzer should be seeded with interesting
      inputs - but not ones that cause an outright crash.

    - Least likely, there is a horrible bug in the fuzzer. If other options
      fail, poke <lcamtuf@coredump.cx> for troubleshooting tips.

[-] PROGRAM ABORT : Test case 'id_000000' results in a crash
         Location : perform_dry_run(), C:\dev\github\winafl\afl-fuzz.c:3321


C:\dev\afl>
```



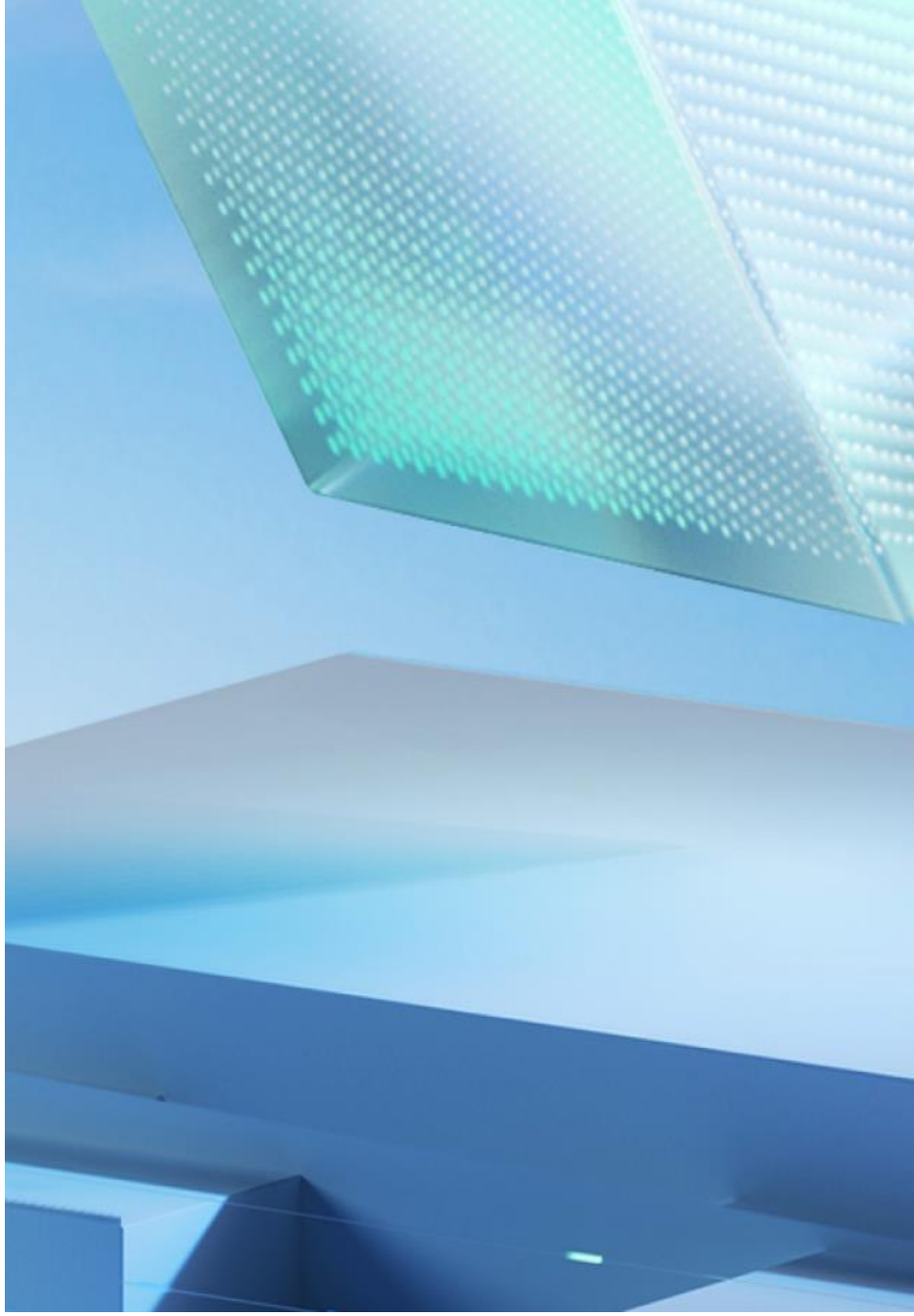
AFL False Start

```
                WinAFL 1.17 based on AFL 2.43b (FuzzX509.exe)

+- process timing ------------------------------------+- overall results ----+
|        run time : 0 days, 0 hrs, 0 min, 31 sec      |  cycles done : 0     |
|   last new path : 0 days, 0 hrs, 0 min, 1 sec       |  total paths : 201   |
| last uniq crash : none seen yet                     | uniq crashes : 0     |
|  last uniq hang : none seen yet                     |   uniq hangs : 0     |
+- cycle progress --------------------+- map coverage -+----------------------+
|  now processing : 2 (1.00%)         |    map density : 0.16% / 3.65%         |
| paths timed out : 0 (0.00%)         | count coverage : 1.33 bits/tuple       |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : havoc                 | favored paths : 135 (67.16%)           |
| stage execs : 9570/32.8k (29.21%)   |  new edges on : 179 (89.05%)           |
| total execs : 52.0k                 | total crashes : 0 (0 unique)           |
|  exec speed : 1637/sec              |  total tmouts : 0 (0 unique)           |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 8/344, 4/342, 4/338                   |    levels : 3          |
|  byte flips : 0/43, 0/41, 0/37                      |   pending : 200        |
| arithmetics : 33/2406, 0/1509, 0/21                 |  pend fav : 135        |
|  known ints : 2/179, 1/1219, 2/1472                 | own finds : 200        |
|  dictionary : 0/0, 0/0, 0/0                         |  imported : n/a        |
|       havoc : 137/32.8k, 0/0                        | stability : 82.85%     |
|        trim : 44.87%/18, 0.00%                      +------------------------+
+-----------------------------------------------------+   [cpu000001:   6%]
```



Timestamp Decoding

```
0:000> wt -i kernelbase -i kernel32 -i verifier -i ucrtbase -i vcruntime140 -i ntdll
   27     0 [  0] CRYPT32!CryptDecodeObjectEx
    5     0 [  1]   CRYPT32!LoadRegFunc
   76 24702 [  0] CRYPT32!CryptDecodeObjectEx
   16     0 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   23   886 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   54     0 [  3]       MSASN1!ASN1_Decode
   14     0 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   77     0 [  5]           MSASN1!ASN1BERDecExplicitTag
   13     0 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   14     0 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   10     0 [  7]               CRYPT32!ASN1DecRealloc_Elements
   81  2474 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   60  2909 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   44  3213 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   95  3301 [  3]       MSASN1!ASN1_Decode
   35  4282 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   32     0 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   29     0 [  5]           CRYPT32!Asn1X509GetPKIFreeText
   45   457 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   80  4367 [  3]       CRYPT32!PkiAsn1AllocStructInfoEx
   42  8729 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   30   116 [  3]       MSASN1!ASN1_FreeDecoded
   51  8875 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   18  8926 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   94 33648 [  0] CRYPT32!CryptDecodeObjectEx
```

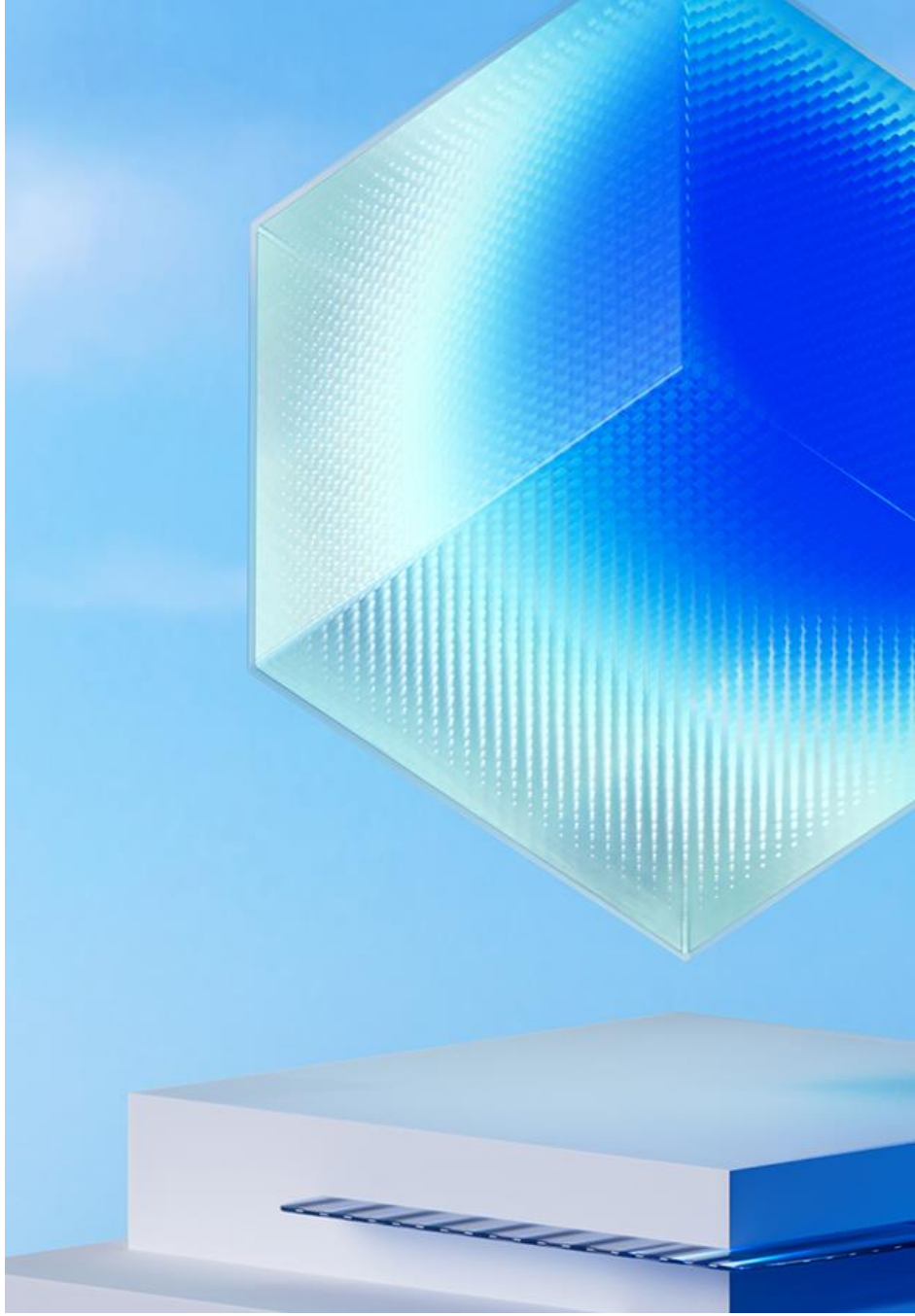


FIELD EFFECT

Strings are Hard

Decompile: Asn1X509GetPKIFreeText - (crypt32.dll)

```
/* void __cdecl Asn1X509GetPKIFreeText(struct PKIFreeText * __ptr64,unsigned long,struct _CRYPT_TIMESTAMP_RESPONSE *
   __ptr64,unsigned char * __ptr64 * __ptr64,long * __ptr64) */

void __cdecl
Asn1X509GetPKIFreeText
          (PKIFreeText *PkiFreeText,ulong param_2,_CRYPT_TIMESTAMP_RESPONSE *TimestampResponse,uchar **OutputBuffer,
          long *BytesLeft)

{
  LPWSTR *currentOutputString;
  int bytesLeft;
  STRING_ENTRY *stringEntry;
  DWORD freeTextCount;
  DWORD roundedStringSize;
  DWORD stringSize;
  LPWSTR outputString;

  freeTextCount = PkiFreeText->StringCount;
  bytesLeft = *BytesLeft + freeTextCount * -0x10;
  *BytesLeft = bytesLeft;
  if (bytesLeft < 0) {
    currentOutputString = (LPWSTR *)0x0;
  }
  else {
    currentOutputString = (LPWSTR *)*OutputBuffer;
    TimestampResponse->cFreeText = freeTextCount;
    *OutputBuffer = (uchar *)((longlong)(int)(freeTextCount * 0x10) + (longlong)currentOutputString);
    TimestampResponse->rgFreeText = currentOutputString;
  }
  stringEntry = (STRING_ENTRY *)PkiFreeText->StringArray;
  for (; freeTextCount != 0; freeTextCount = freeTextCount - 1) {
    stringSize = stringEntry->StringSize;
    roundedStringSize = stringSize + 9 & 0xfffffff8;
    bytesLeft = bytesLeft - roundedStringSize;
    *BytesLeft = bytesLeft;
    if (-1 < bytesLeft) {
      outputString = (LPWSTR)*OutputBuffer;
      *currentOutputString = outputString;
      if (stringSize != 0) {
        memcpy(outputString,stringEntry->StringPointer,(ulonglong)stringSize);
      }
      (*currentOutputString)[stringSize] = L'\0';
      *OutputBuffer = (uchar *)((longlong)(int)roundedStringSize + (longlong)outputString);
    }
    currentOutputString = currentOutputString + 1;
    stringEntry = stringEntry + 1;
  }
  return;
}
```



FIELD EFFECT

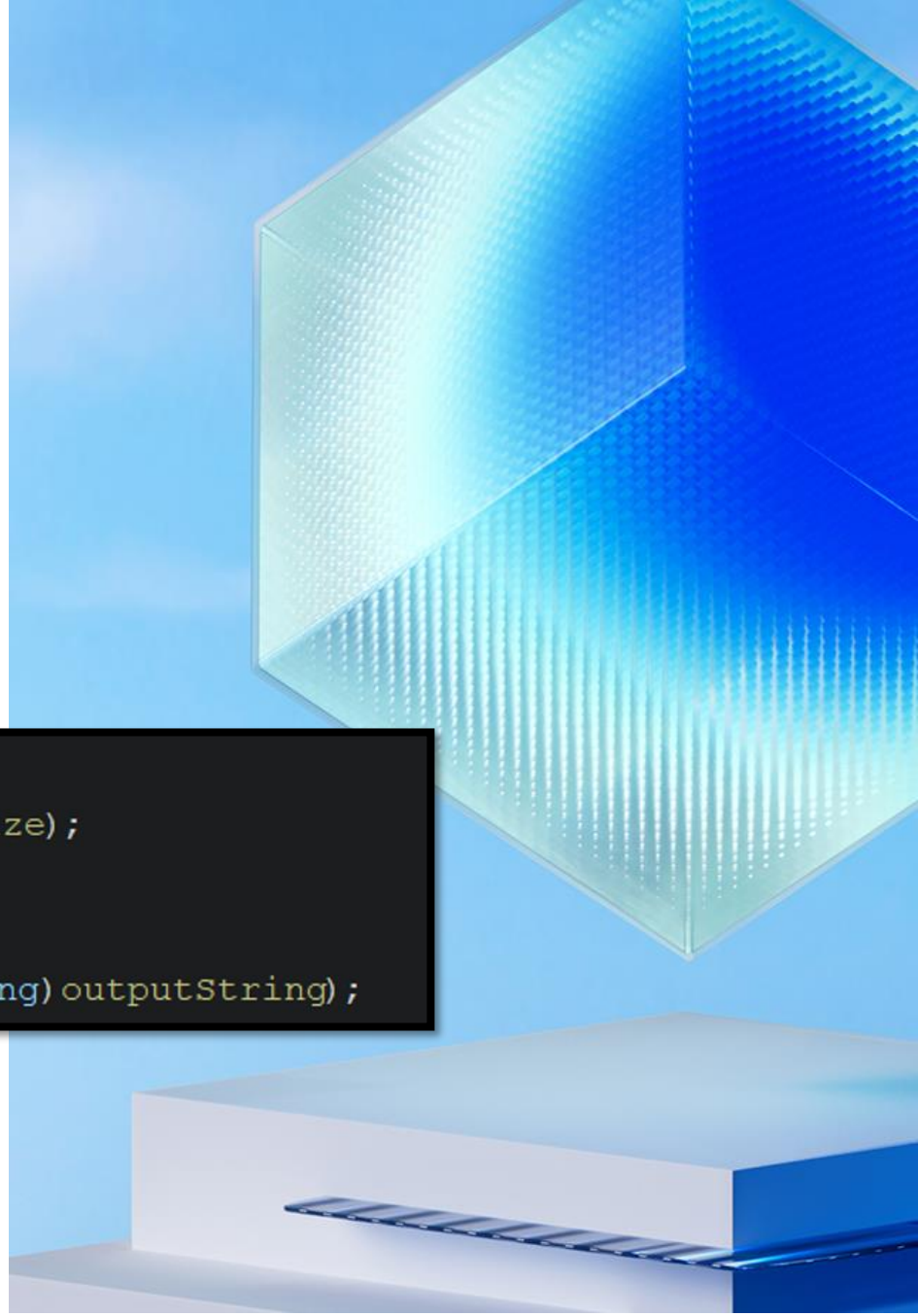
Strings are Hard

Decompile: Asn1X509GetPKIFreeText - (crypt32.dll)

```
/* void __cdecl Asn1X509GetPKIFreeText(struct PKIFreeText * __ptr64,unsigned long,struct _CRYPT_TIMESTAMP_RESPONSE *
   __ptr64,unsigned char * __ptr64 * __ptr64,long * __ptr64) */

void __cdecl
Asn1X509GetPKIFreeText
          (PKIFreeText *PkiFreeText,ulong param_2,_CRYPT_TIMESTAMP_RESPONSE *TimestampResponse,uchar **OutputBuffer,
          long *BytesLeft)

{
  LPWSTR *currentOutputString;
  int bytesLeft;
  STRING_ENTRY *stringEntry;
  DWORD freeTextCount;
  DWORD roundedStringSize;
  DWORD stringSize;
  LPWSTR outputString;

  freeTextCount = PkiFreeText->StringCount;
```

```
if (stringSize != 0) {
  memcpy(outputString,stringEntry->StringPointer,(ulonglong)stringSize);
}
(*currentOutputString)[stringSize] = L'\0';
*OutputBuffer = (uchar *)((longlong)(int)roundedStringSize + (longlong)outputString);
```

```
    bytesLeft = bytesLeft - roundedStringSize;
    *BytesLeft = bytesLeft;
    if (-1 < bytesLeft) {
      outputString = (LPWSTR)*OutputBuffer;
      *currentOutputString = outputString;
      if (stringSize != 0) {
        memcpy(outputString,stringEntry->StringPointer,(ulonglong)stringSize);
      }
      (*currentOutputString)[stringSize] = L'\0';
      *OutputBuffer = (uchar *)((longlong)(int)roundedStringSize + (longlong)outputString);
    }
    currentOutputString = currentOutputString + 1;
    stringEntry = stringEntry + 1;
  }
  return;
}
```



FIELD EFFECT

Where Is This Used

The **CryptRetrieveTimeStamp** function encodes a time stamp request and retrieves the time stamp token from a location specified by a URL to a Time Stamping Authority (TSA).

Syntax

C++ Copy

```
BOOL CryptRetrieveTimeStamp(
  [in]            LPCWSTR                    wszUrl,
                  DWORD                      dwRetrievalFlags,
                  DWORD                      dwTimeout,
  [in]            LPCSTR                     pszHashId,
  [in, optional]  const CRYPT_TIMESTAMP_PARA *pPara,
  [in]            const BYTE                 *pbData,
                  DWORD                      cbData,
  [out]           PCRYPT_TIMESTAMP_CONTEXT   *ppTsContext,
  [out, optional] PCCERT_CONTEXT             *ppTsSigner,
  [out, optional] HCERTSTORE                 *phStore
);
```

Parameters

`[in] wszUrl`

A pointer to a null-terminated wide character string that contains the URL of the TSA to which to send the request.

Signtool PoC

SignTool

Article • 03/28/2022 • 6 contributors

Feedback

In this article

- Partial list of operations, options, and arguments
- Remarks
- Examples

SignTool (Signtool.exe) is a command-line CryptoAPI tool that digitally-signs files, verifies signatures in files, and time stamps files.

command

Copy

```
SignTool [Operation] [Options] [FileName ...]
```

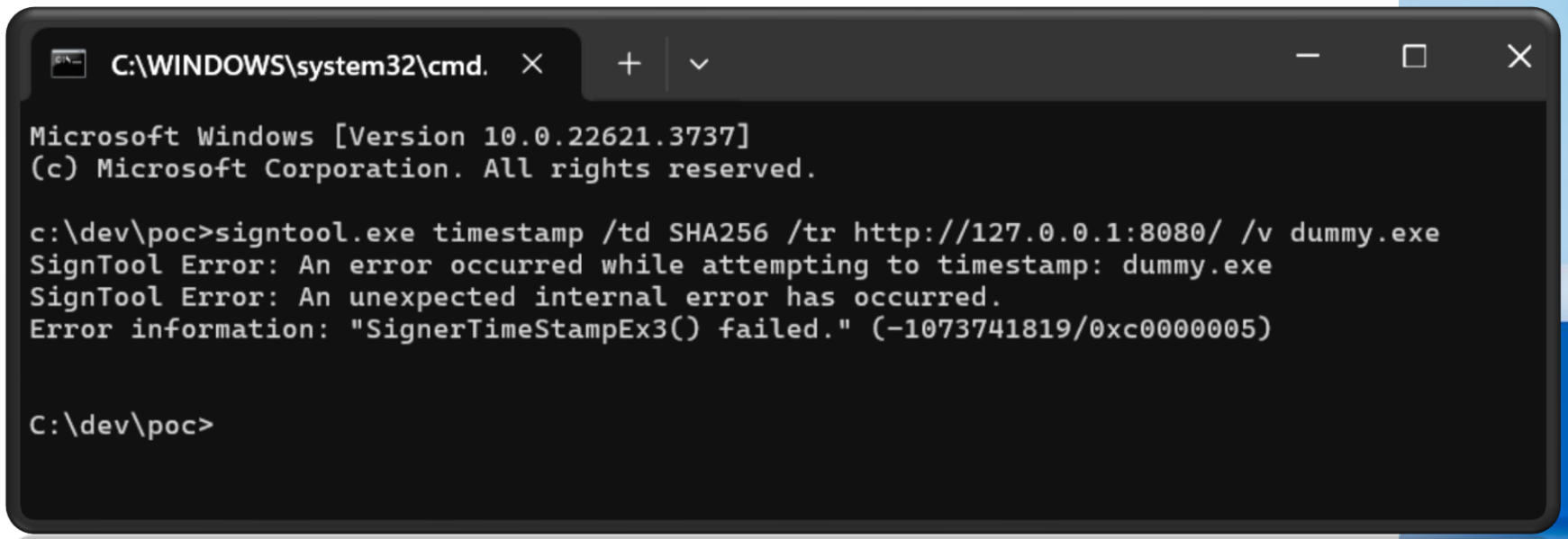
The following command signs and time stamps the file:

```
SignTool sign /f MyCert.pfx /t http://timestamp.digicert.com MyControl.exe
```

Note

For information about time stamping a file after it has already been signed, see Adding

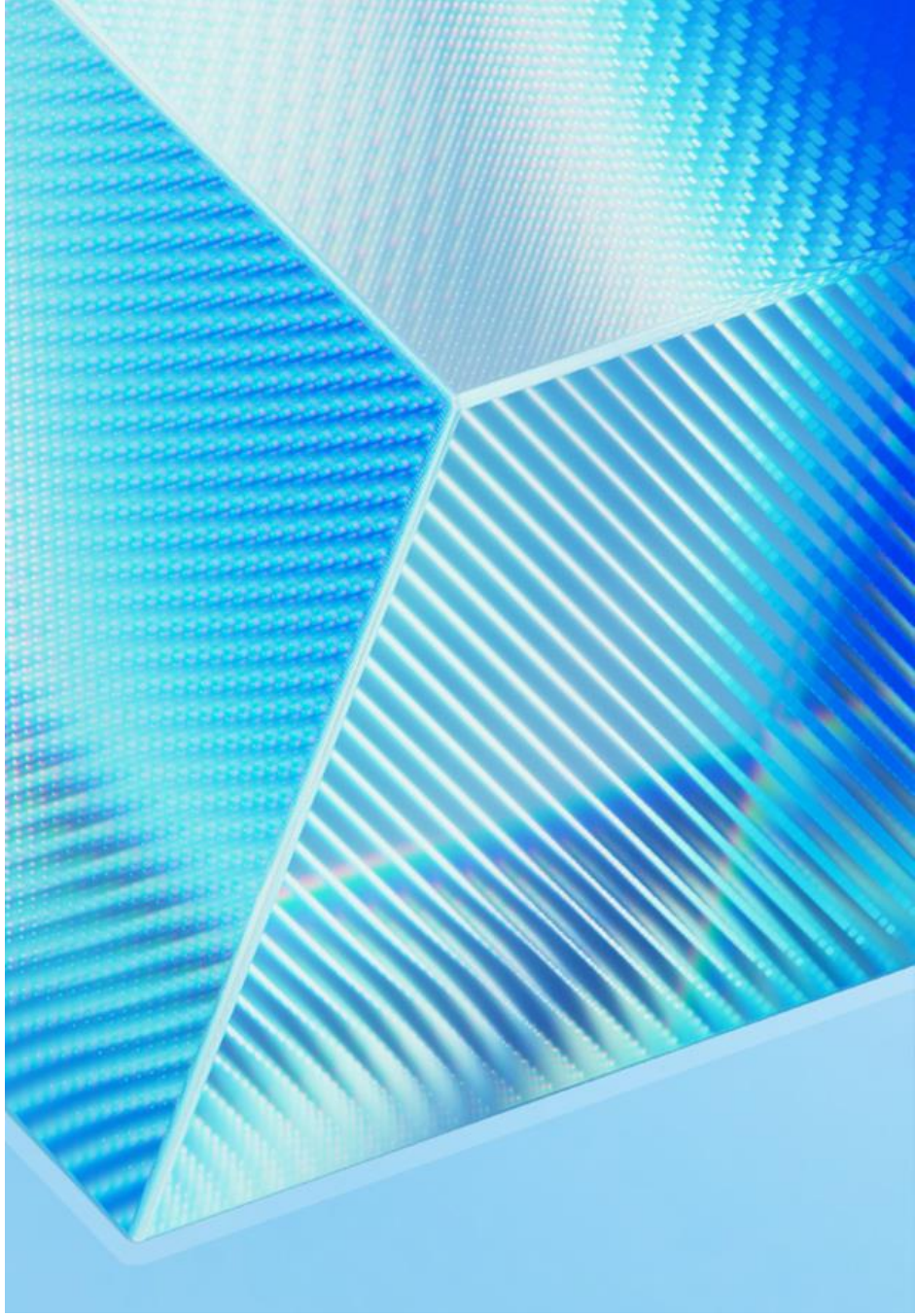
CVE-2024-30020



FIELD EFFECT

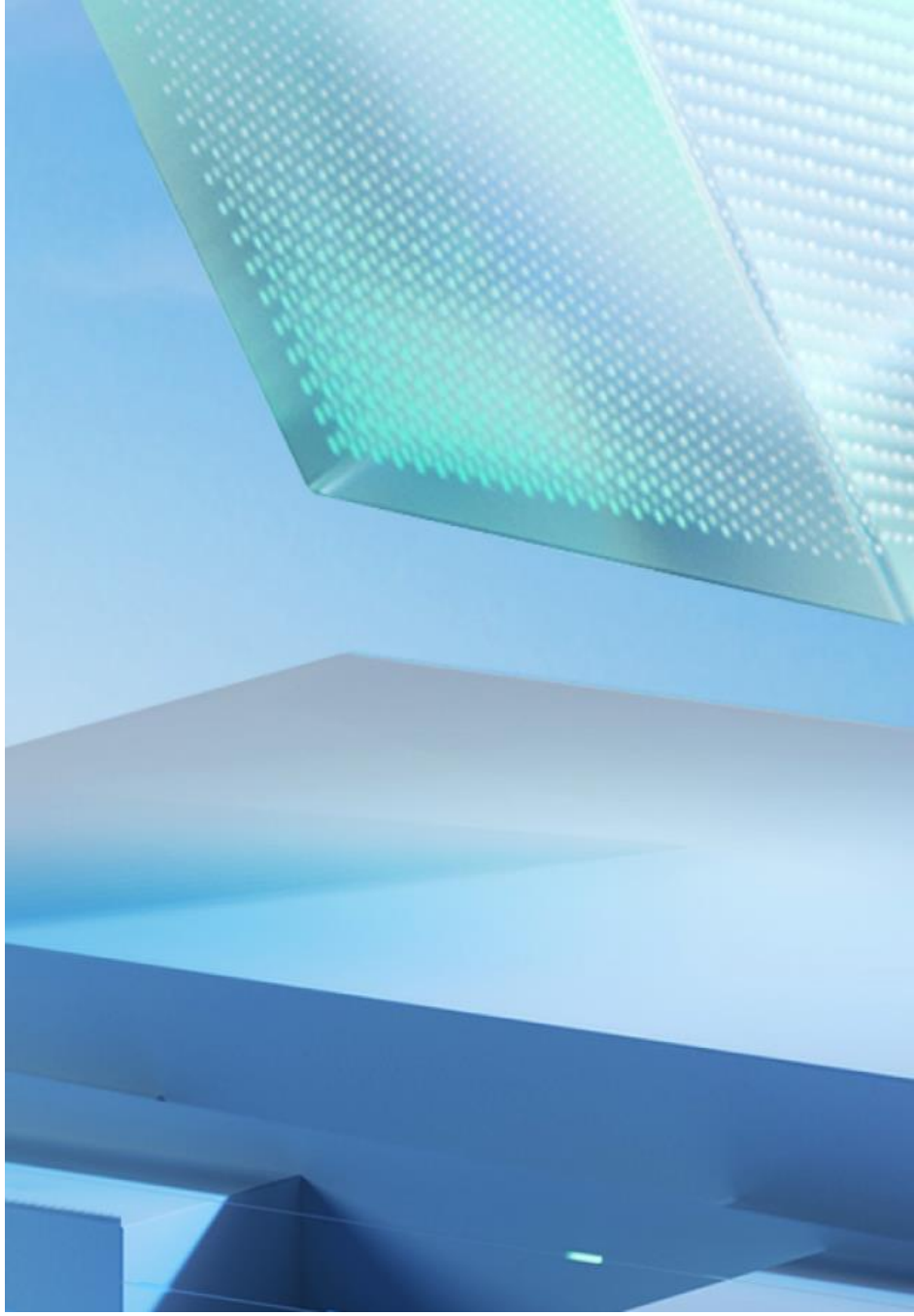
UserNotice Fuzzing

```
+----------------------------------------------------+   [cpu000001:   6%]
               WinAFL 1.17 based on AFL 2.43b (FuzzX509.exe)

+- process timing ------------------------------------+- overall results -----+
|        run time : 0 days, 0 hrs, 31 min, 3 sec      |  cycles done : 0      |
|   last new path : 0 days, 0 hrs, 0 min, 0 sec       |  total paths : 947    |
| last uniq crash : none seen yet                     | uniq crashes : 0      |
|  last uniq hang : none seen yet                     |   uniq hangs : 0      |
+- cycle progress --------------------+- map coverage -+-----------------------+
|  now processing : 795 (83.95%)      |    map density : 0.36% / 7.71%         |
| paths timed out : 0 (0.00%)         | count coverage : 1.48 bits/tuple       |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : havoc                 | favored paths : 592 (62.51%)           |
| stage execs : 12.2k/24.6k (49.48%)  |  new edges on : 690 (72.86%)           |
| total execs : 2.38M                 | total crashes : 0 (0 unique)           |
|  exec speed : 1298/sec              |  total tmouts : 0 (0 unique)           |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 115/46.9k, 45/46.5k, 31/45.5k         |    levels : 6          |
|  byte flips : 1/5863, 3/5409, 2/4526                |   pending : 494        |
| arithmetics : 183/327k, 4/167k, 4/42.0k             |  pend fav : 107        |
|  known ints : 33/24.6k, 21/148k, 17/163k            | own finds : 946        |
|  dictionary : 0/0, 0/0, 6/16.8k                     |  imported : n/a        |
|       havoc : 478/1.31M, 0/0                        | stability : 89.25%     |
|        trim : 44.21%/2090, 0.00%                    +------------------------+
+----------------------------------------------------+   [cpu000001:   4%]
```



Timestamp Decoding II

```
0:000> wt -i kernelbase -i kernel32 -i verifier -i ucrtbase -i vcruntime140 -i ntdll
   27     0 [  0] CRYPT32!CryptDecodeObjectEx
    5     0 [  1]   CRYPT32!LoadRegFunc
   76 24702 [  0] CRYPT32!CryptDecodeObjectEx
   16     0 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   23   886 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   54     0 [  3]       MSASN1!ASN1_Decode
   14     0 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   77     0 [  5]           MSASN1!ASN1BERDecExplicitTag
   13     0 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   14     0 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   10     0 [  7]               CRYPT32!ASN1DecRealloc_Elements
   81  2474 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   60  2909 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   44  3213 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   95  3301 [  3]       MSASN1!ASN1_Decode
   35  4282 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   32     0 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   29     0 [  5]           CRYPT32!Asn1X509GetPKIFreeText
   45   457 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   80  4367 [  3]       CRYPT32!PkiAsn1AllocStructInfoEx
   42  8729 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   30   116 [  3]       MSASN1!ASN1_FreeDecoded
   51  8875 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   18  8926 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   94 33648 [  0] CRYPT32!CryptDecodeObjectEx
```



FIELD EFFECT

CryptDecodeObjectEx Internals

```C++
BOOL CryptDecodeObjectEx(
  [in]      DWORD              dwCertEncodingType,
  [in]      LPCSTR             lpszStructType,
  [in]      const BYTE         *pbEncoded,
  [in]      DWORD              cbEncoded,
  [in]      DWORD              dwFlags,
  [in]      PCRYPT_DECODE_PARA pDecodePara,
  [out]     void               *pvStructInfo,
  [in, out] DWORD              *pcbStructInfo
);
```

DER Data

ASN1Dec_* functions

```
A time-stamping response is as follows:

TimeStampResp ::= SEQUENCE  {
   status                  PKIStatusInfo,
   timeStampToken          TimeStampToken     OPTIONAL  }

The status is based on the definition of status in section 3.2.3
of [RFC2510] as follows:

PKIStatusInfo ::= SEQUENCE {
   status        PKIStatus,
   statusString  PKIFreeText     OPTIONAL,
   failInfo      PKIFailureInfo  OPTIONAL  }
```

ASN1...DecodeEx functions

```C++
typedef struct _CRYPT_TIMESTAMP_RESPONSE {
  DWORD          dwStatus;
  DWORD          cFreeText;
  LPWSTR         *rgFreeText;
  CRYPT_BIT_BLOB FailureInfo;
  CRYPT_DER_BLOB ContentInfo;
} CRYPT_TIMESTAMP_RESPONSE, *PCRYPT_TIMESTAMP_RESPONSE;
```

FIELD EFFECT

ASN1Dec_PKIFreeText

Decompile: ASN1Dec_PKIFreeText - (crypt32.dll)

```
1
2  bool ASN1Dec_PKIFreeText(ASN1decoding_s *Decoder,DWORD Tag,astruct_2 *Asn1FreeText)
3
4  {
5    int success;
6    ANSI_STRING *localBuffer;
7    uint bufferSize;
8    undefined4 tag [2];
9    ASN1decoding_s localDecoder;
10   void **contents;
11   uint stringCount;
12
13   bufferSize = 0;
14   _localDecoder = 0;
15   contents = (void **)0x0;
16   tag[0] = 0;
17   success = ASN1BERDecExplicitTag(Decoder,0x10,&localDecoder,&contents);
18   if (success != 0) {
19     Asn1FreeText->StringCount = 0;
20     Asn1FreeText->StringArray = (ANSI_STRING *)0x0;
21     do {
22       success = ASN1BERDecNotEndOfContents(_localDecoder,contents);
23       if (success == 0) {
24         success = ASN1BERDecEndOfContents(Decoder,_localDecoder,contents);
25         return success != 0;
26       }
27       success = ASN1BERDecPeekTag(_localDecoder,tag);
28       if (success == 0) {
29         return false;
30       }
31       if (bufferSize <= Asn1FreeText->StringCount) {
32         if (bufferSize == 0) {
33           bufferSize = 0x10;
34         }
35         else {
36           bufferSize = bufferSize * 2;
37         }
38         localBuffer = (ANSI_STRING *)ASN1DecRealloc(_localDecoder,Asn1FreeText->StringArray,bufferSize << 4);
39         if (localBuffer == (ANSI_STRING *)0x0) {
40           return false;
41         }
42         Asn1FreeText->StringArray = localBuffer;
43       }
44       stringCount = Asn1FreeText->StringCount;
45       Asn1FreeText->StringCount = stringCount + 1;
46       success = ASN1BERDecUTF8String(_localDecoder,0xc,Asn1FreeText->StringArray + stringCount);
47     } while (success != 0);
48   }
49   return false;
50 }
```

```
PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
    -- text encoded as UTF-8 String [RFC3629] (note: each
    -- UTF8String MAY include an [RFC3066] language tag
    -- to indicate the language of the contained text
    -- see [RFC2482] for details)
```

FIELD EFFECT

ASN1Dec_PKIFreeText

```
PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
    -- text encoded as UTF-8 String [RFC3629] (note: each
    -- UTF8String MAY include an [RFC3066] language tag
    -- to indicate the language of the contained text
    -- see [RFC2482] for details)
```

Decompile: ASN1Dec_PKIFreeText - (crypt32.dll)

```
1
2  bool ASN1Dec_PKIFreeText(ASN1decoding_s *Decoder,DWORD Tag,astruct_2 *Asn1FreeText)
3
4  {
5    int success;
6    ANSI_STRING *localBuffer;
7    uint bufferSize;
8    undefined4 tag [2];
9    ASN1decoding_s localDecoder;
10   void **contents;
```

```
if (bufferSize <= Asn1FreeText->StringCount) {
  if (bufferSize == 0) {
    bufferSize = 0x10;
  }
  else {
    bufferSize = bufferSize * 2;
  }
  localBuffer = (ANSI_STRING *)ASN1DecRealloc(_localDecoder,Asn1FreeText->StringArray,bufferSize << 4);
  if (localBuffer == (ANSI_STRING *)0x0) {
    return false;
  }
  Asn1FreeText->StringArray = localBuffer;
}
stringCount = Asn1FreeText->StringCount;
Asn1FreeText->StringCount = stringCount + 1;
success = ASN1BERDecUTF8String(_localDecoder,0xc,Asn1FreeText->StringArray + stringCount);
```

```
49   return false;
50 }
```

FIELD EFFECT

ASN1Dec_PKIFreeText

Decompile: ASN1Dec_PKIFreeText - (crypt32.dll)

```
1
2 bool ASN1Dec_PKIFreeText(ASN1decoding_s *Decoder,DWORD Tag,astruct_2 *Asn1FreeText)
3
4 {
5   int success;
6   ANSI_STRING *localBuffer;
7   uint bufferSize;
8   undefined4 tag [2];
9   ASN1decoding_s localDecoder;
10  void **contents;
```

```
if (bufferSize <= Asn1FreeText->StringCount) {
  if (bufferSize == 0) {
    bufferSize = 0x10;
  }
  else {
    bufferSize = bufferSize * 2;
  }
  localBuffer = (ANSI_STRING *)ASN1DecRealloc(_localDecoder,Asn1FreeText->StringArray,bufferSize << 4);
  if (localBuffer == (ANSI_STRING *)0x0) {
    return false;
  }
  Asn1FreeText->StringArray = localBuffer;
}
stringCount = Asn1FreeText->StringCount;
Asn1FreeText->StringCount = stringCount + 1;
success = ASN1BERDecUTF8String(_localDecoder,0xc,Asn1FreeText->StringArray + stringCount);
```

```
49  return false;
50 }
```

FIELD EFFECT

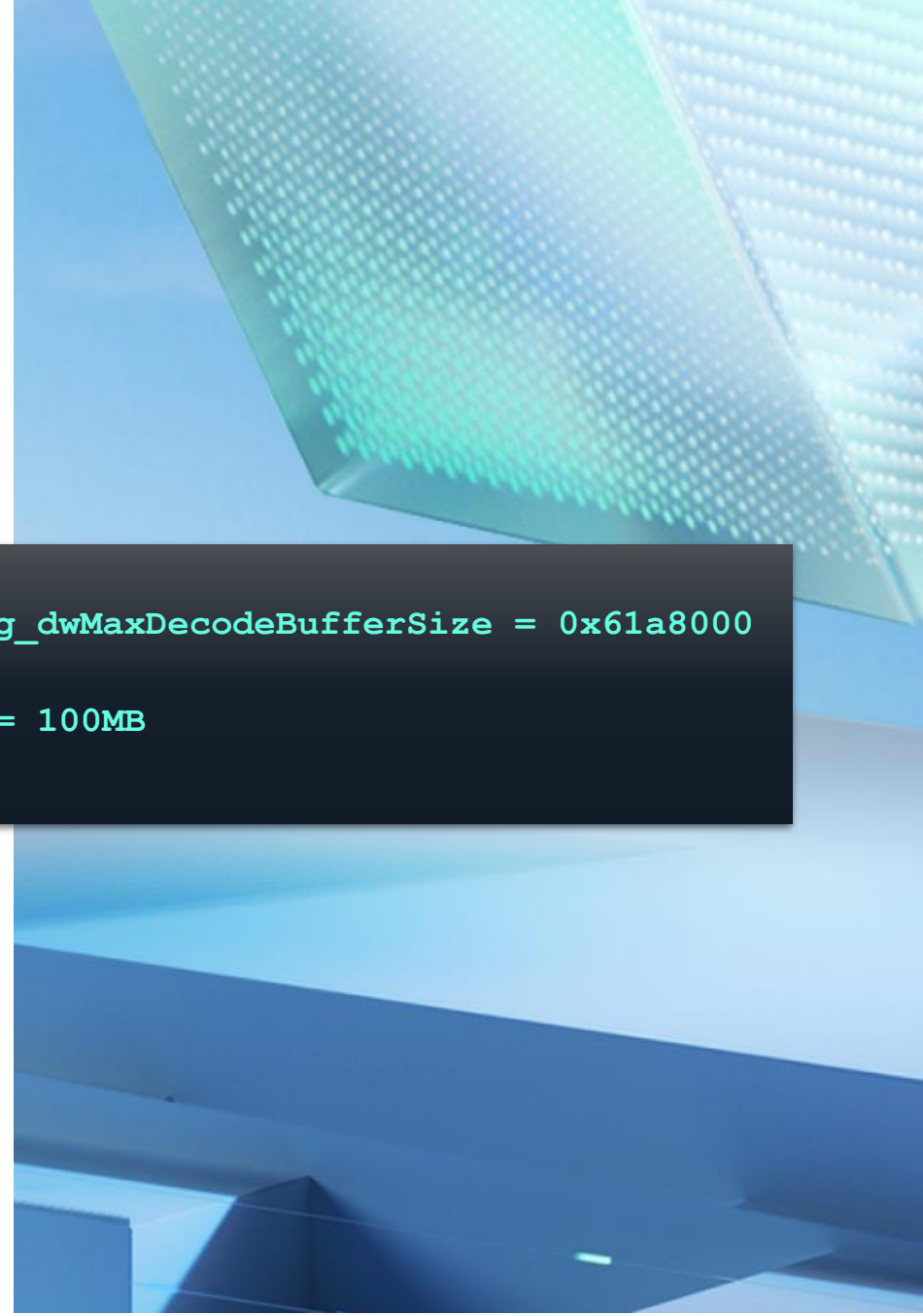
PoC Attempt

```
void poc()
{
    CRYPT_TIMESTAMP_RESPONSE timestampResponse = { 0 };
    DWORD arraySize = 0;
    void* encodedResponse = NULL;
    DWORD encodedSize = 0;
    CRYPT_TIMESTAMP_RESPONSE* decodedResponse = NULL;
    DWORD decodedSize = 0;

    timestampResponse.cFreeText = 0xc000000;
    arraySize = timestampResponse.cFreeText * sizeof( LPWSTR* );
    timestampResponse.rgFreeText = (LPWSTR*) malloc( arraySize );
    if( NULL != timestampResponse.rgFreeText )
        memset( timestampResponse.rgFreeText, 0, arraySize );

    if( !CryptEncodeObjectEx(
        X509_ASN_ENCODING | PKCS_7_ASN_ENCODING,
        TIMESTAMP_RESPONSE,
        (void*) &timestampResponse,
        CRYPT_ENCODE_ALLOC_FLAG,
        NULL,
        &encodedResponse,
        &encodedSize ) )
    {
        printf( "CryptEncodeObjectEx() failed: 0x%08x\n", GetLastError() );
    }
    else
    {
        if( !CryptDecodeObjectEx(
            X509_ASN_ENCODING | PKCS_7_ASN_ENCODING,
            TIMESTAMP_RESPONSE,
            encodedResponse,
            encodedSize,
            CRYPT_DECODE_ALLOC_FLAG,
            NULL,
            &decodedResponse,
            &decodedSize ) )
        {
            printf( "CryptDecodeObjectEx() failed: 0x%08x\n", GetLastError() );
        }
    }
}
```

```
Count = 0xc000000
Count * sizeof( UTF8_STRING ) = 0xc000000 * 0x10
      = 0xc0000000

...
alloc = 0x1000000 * 0x10
alloc = 0x2000000 * 0x10
alloc = 0x4000000 * 0x10
alloc = 0x8000000 * 0x10
alloc = 0x10000000 * 0x10 ***
```

Size Checks

```
0:000> wt -i kernelbase -i kernel32 -i verifier -i ucrtbase -i vcruntime140 -i ntdll
00007ff7`1f3a13d0
   27     0 [  0] CRYPT32!CryptDecodeObjectEx
    5     0 [  1]   CRYPT32!LoadRegFunc
   76 24702 [  0] CRYPT32!CryptDecodeObjectEx
   16     0 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   23   886 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   54     0 [  3]       MSASN1!ASN1_Decode
   14     0 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   77     0 [  5]           MSASN1!ASN1BERDecExplicitTag
   13     0 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   14     0 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   10     0 [  7]               CRYPT32!ASN1DecRealloc_Elements
   81  2474 [  6]             CRYPT32!ASN1Dec_PKIFreeText
   60  2909 [  5]           CRYPT32!ASN1Dec_PKIStatusInfo
   44  3213 [  4]         CRYPT32!ASN1Dec_TimeStampResp
   95  3301 [  3]       MSASN1!ASN1_Decode
   35  4282 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   32     0 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   29     0 [  5]           CRYPT32!Asn1X509GetPKIFreeText
   45   457 [  4]         CRYPT32!Asn1TimeStampResponseExCallback
   80  4367 [  3]       CRYPT32!PkiAsn1AllocStructInfoEx
   42  8729 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   30   116 [  3]       MSASN1!ASN1_FreeDecoded
   51  8875 [  2]     CRYPT32!Asn1InfoDecodeAndAllocEx
   18  8926 [  1]   CRYPT32!Asn1TimeStampResponseDecodeEx
   94 33648 [  0] CRYPT32!CryptDecodeObjectEx
```

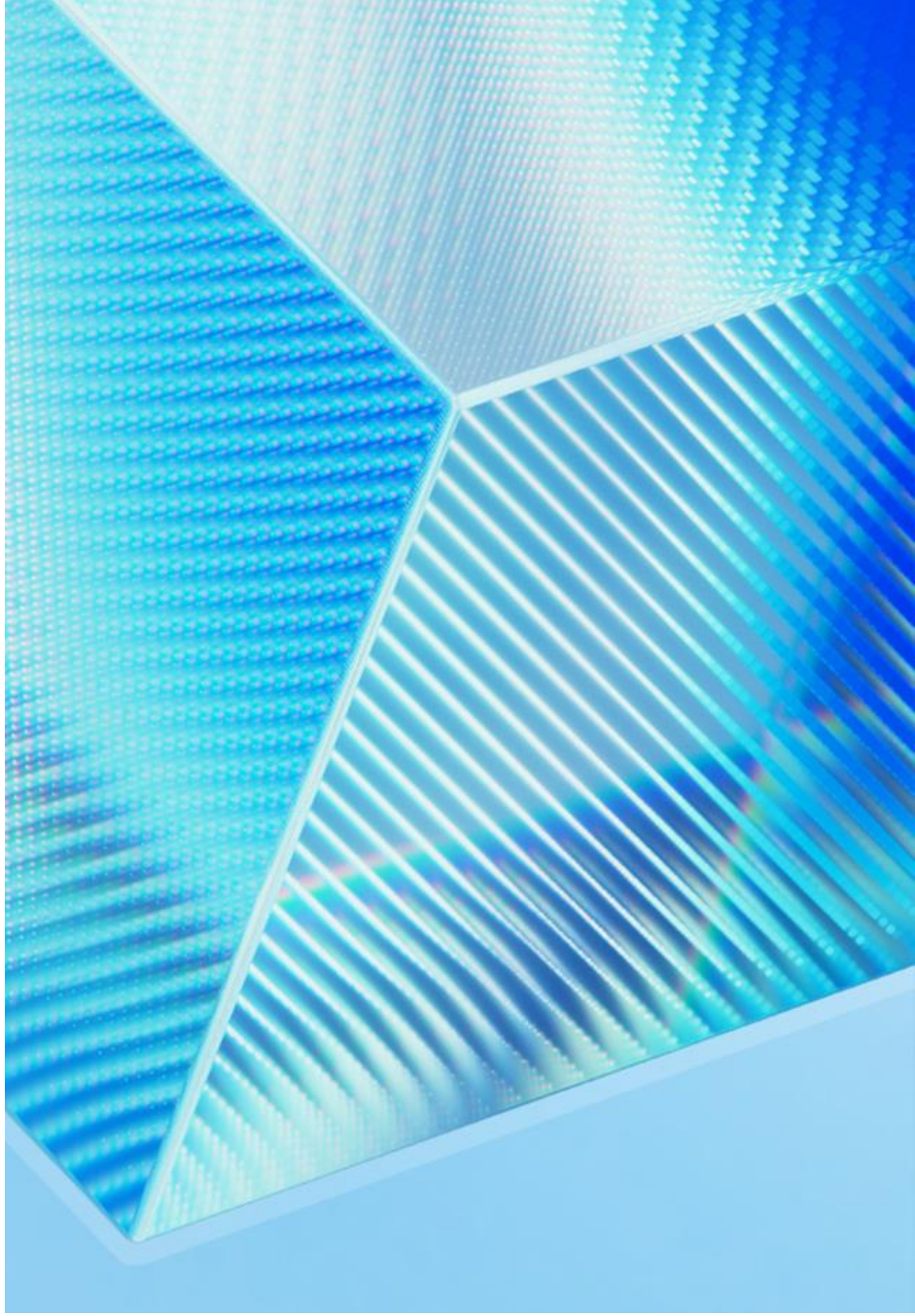


```
g_dwMaxDecodeBufferSize = 0x61a8000

= 100MB
```

FIELD EFFECT

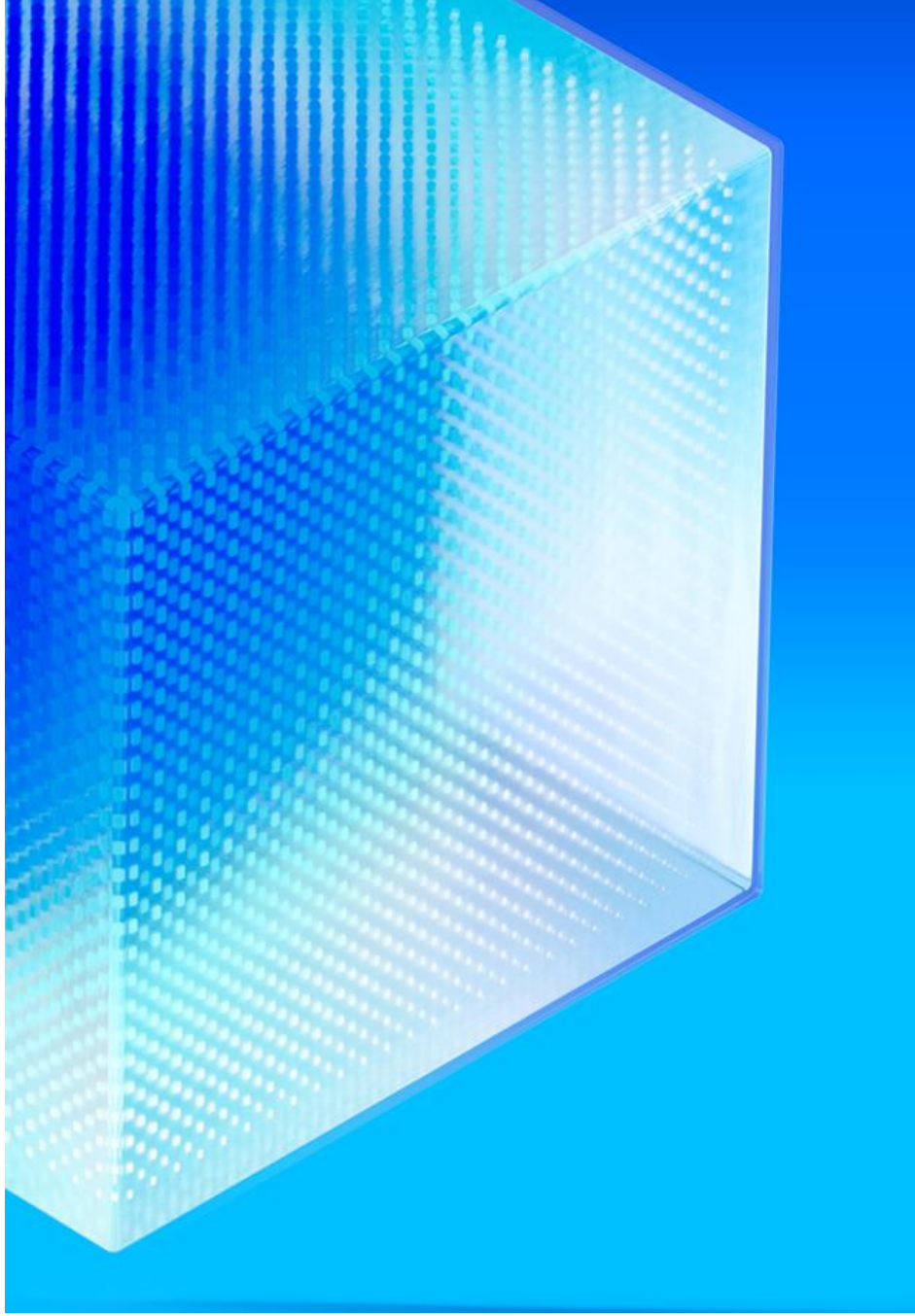
UserNotice Fuzzing

```
+----------------------------------------------------+   [cpu000001:   4%]
               WinAFL 1.17 based on AFL 2.43b (FuzzX509.exe)

+- process timing ------------------------------------+- overall results ----+
|        run time : 0 days, 1 hrs, 14 min, 2 sec      |  cycles done : 4     |
|   last new path : 0 days, 0 hrs, 0 min, 7 sec       |  total paths : 1337  |
| last uniq crash : none seen yet                     | uniq crashes : 0     |
|  last uniq hang : none seen yet                     |   uniq hangs : 0     |
+- cycle progress --------------------+- map coverage -+----------------------+
|  now processing : 1326* (99.18%)    |    map density : 0.25% / 9.17%         |
| paths timed out : 0 (0.00%)         | count coverage : 1.57 bits/tuple       |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : havoc                 | favored paths : 701 (52.43%)           |
| stage execs : 3240/8192 (39.55%)    |  new edges on : 904 (67.61%)           |
| total execs : 5.58M                 | total crashes : 0 (0 unique)           |
|  exec speed : 1207/sec              |  total tmouts : 0 (0 unique)           |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 199/118k, 78/117k, 52/115k            |    levels : 14         |
|  byte flips : 1/14.8k, 4/13.9k, 3/12.1k             |   pending : 414        |
| arithmetics : 285/828k, 7/439k, 5/83.1k             |  pend fav : 2          |
|  known ints : 54/62.8k, 42/394k, 29/449k            | own finds : 1336       |
|  dictionary : 0/0, 0/0, 22/145k                     |  imported : n/a        |
|       havoc : 555/2.77M, 0/0                        | stability : 88.45%     |
|        trim : 34.71%/4739, 0.00%                    +------------------------+
+----------------------------------------------------+   [cpu000001:   3%]
```



Vulnerability Constraints

- Input Buffer Size <= 100MB
- Object Count * Object Size > 0x80000000 (2GB)
- ASN1 Decoding Expansion = 2GB / 100MB
- Need ~20x Expansion
- Encoding Must Be Valid



FIELD EFFECT

ASN1DecRealloc Xref

References to ASN1DecRealloc - 87 locations

Locati...	Label	Code Unit	Context	Function Name
18007d7bf		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_OcspBasicResponseList
18007df71		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_CertificateTrustList
18007e657		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_OcspCerts
18007e9f3		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_Extensions
18007ec76		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_TrustedSubjects
18007ed74		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_Attributes
18007ee83		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_Attributes
18007f307		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_Name
18007f40a		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_Name
18007f57a		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_RelativeDistinguishedName
18007fa63		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_GeneralSubtrees
18007fc6d		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_AuthorityInfoAccess
180080120		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_CRLDistributionPoints
18008053a		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_AltNames
180080b63		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_SeqOfAny
180087cda		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_CertificateToBeSigned
180088196		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_SignerInfoWithBlobs
18008824b		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_SignerInfoWithBlobs
180088570		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_AttributesNC2
180088621		CALL qword ptr [->MSASN1.DLL...	READ	ASN1Dec_AttributesNC2

Filter:

FIELD EFFECT

Validate Exploitability

```
35      if (uVar6 <= *param_3) {
36        if (uVar6 == 0) {
37          uVar6 = 0x10;
38        }
39        else {
40          uVar6 = uVar6 * 2;
41        }
42        puVar5 = *(undefined4 **)(param_3 + 2);
43        lVar4 = ASN1DecRealloc(local_res20,puVar5,uVar6 * 0x98);
44        if (lVar4 == 0) {
45          return false;
46        }
47        *(longlong *)(param_3 + 2) = lVar4;
48      }
49      uVar1 = *param_3;
50      *param_3 = uVar1 + 1;
51      bVar2 = ASN1Dec_RecipientEncryptedKey
52                        (local_res20,puVar5,(undefined2 *)((ulonglong)uVar1 * 0x98 + *(longlong *)(param_3 + 2)));
53    } while ((int)CONCAT71(extraout_var,bVar2) != 0);
54  }
55  return false;
56}
```

FIELD EFFECT

ASN1Dec_CRLDistributionPoints()

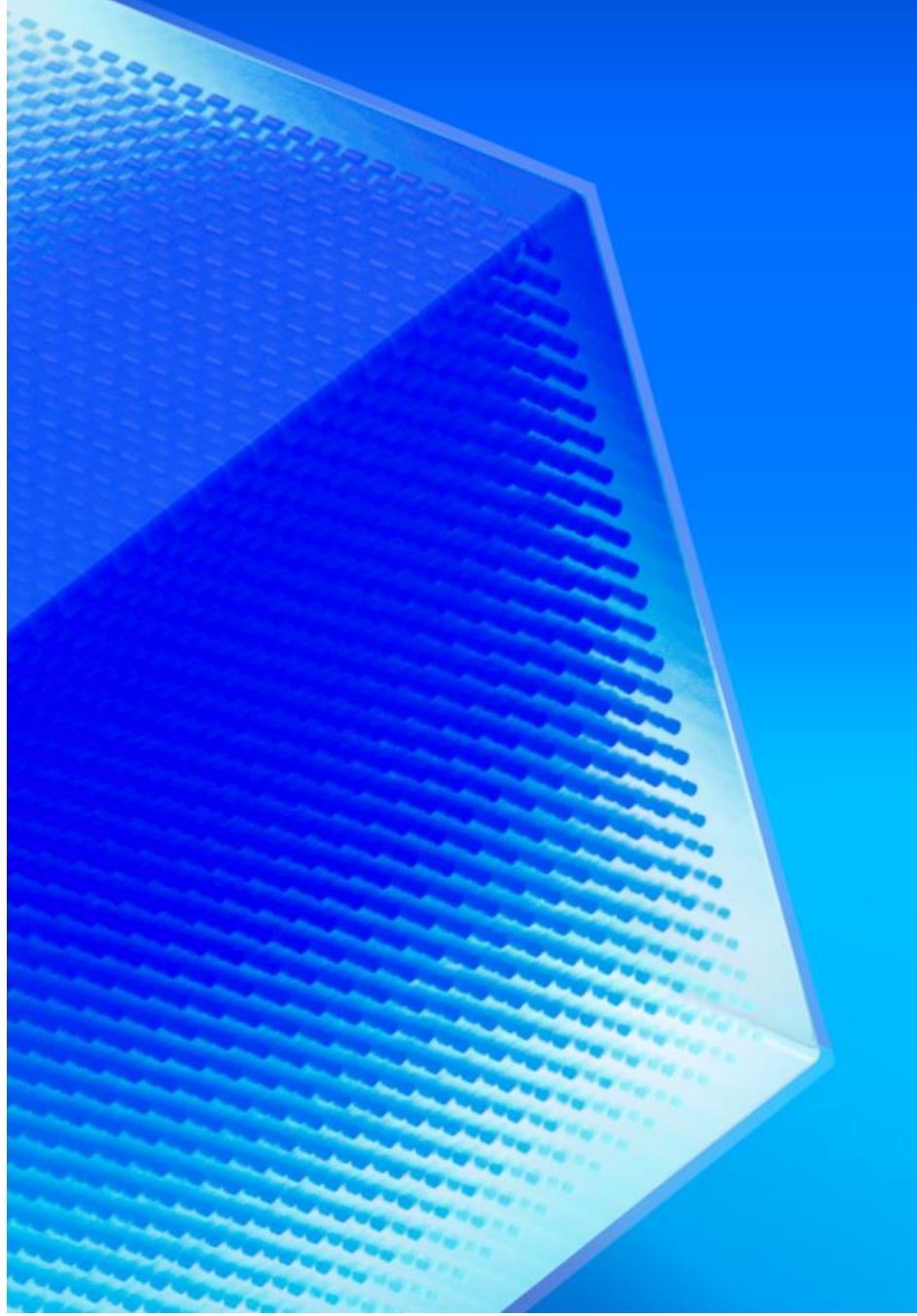
```
48      if (maxObjectCount <= CrlDistPoints->distPointCount) {
49        if (maxObjectCount != 0) {
50          initialObjectCount = maxObjectCount * 2;
51        }
52        distPointeArray = ASN1DecRealloc(local_68,CrlDistPoints->distPointArray,initialObjectCount << 6);
53        if (distPointeArray == 0) {
54          return false;
55        }
56        CrlDistPoints->distPointArray = distPointeArray;
57        maxObjectCount = initialObjectCount;
58      }
```

```
id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::=  { id-ce 31 }

CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
     distributionPoint       [0]     DistributionPointName OPTIONAL,
     reasons                 [1]     ReasonFlags OPTIONAL,
     cRLIssuer               [2]     GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
     fullName                [0]     GeneralNames,
     nameRelativeToCRLIssuer [1]     RelativeDistinguishedName }
```

```
Object Size = (1 << 6) = 0x40

Input Size = 2 Bytes

Ratio = 0x20!!
```

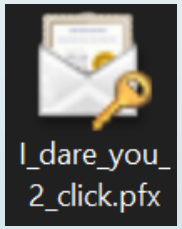
Vulnerability Options

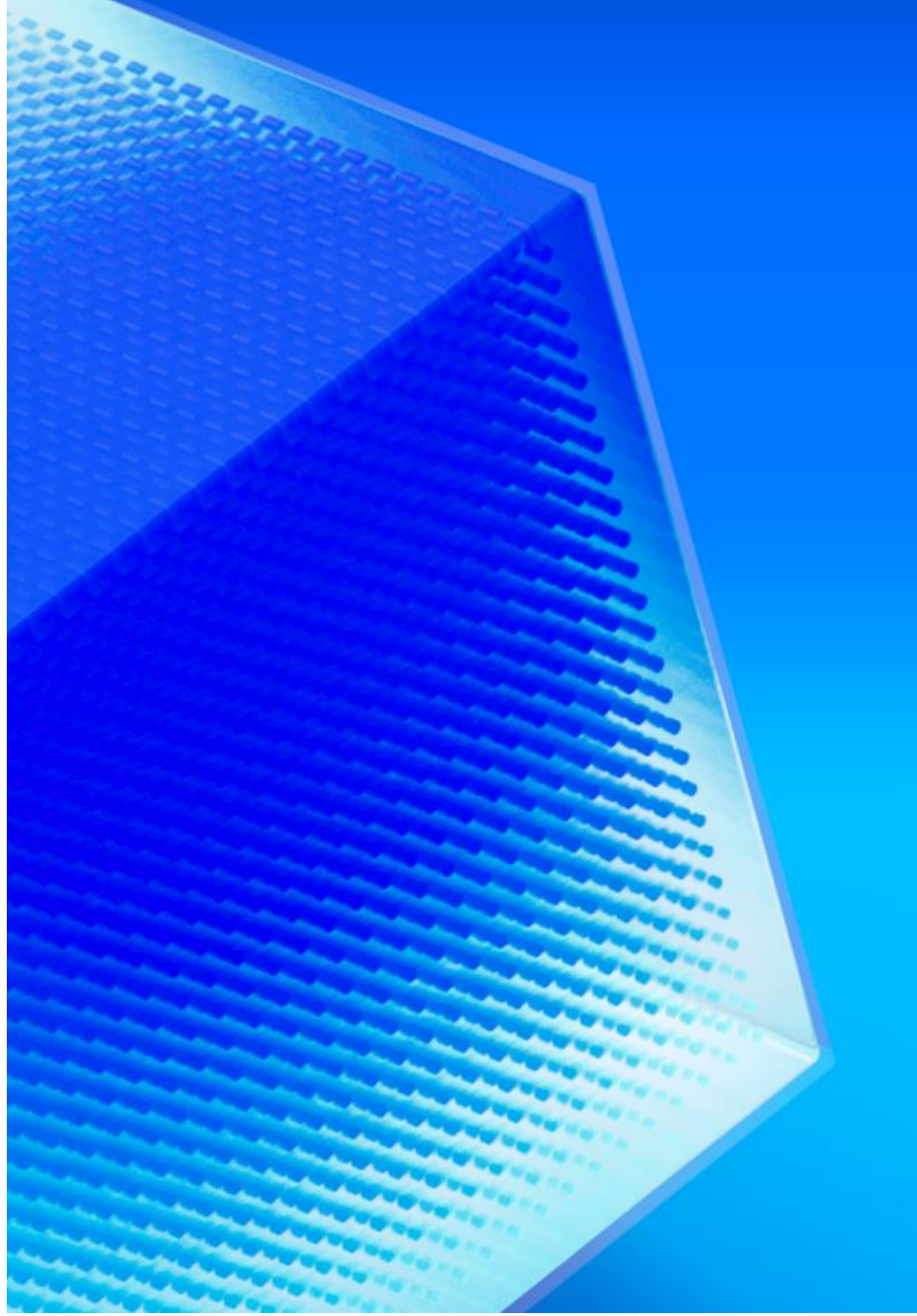
- `szOID_CRL_DIST_POINTS = "2.5.29.31"`
-
-
-



Vulnerability Options

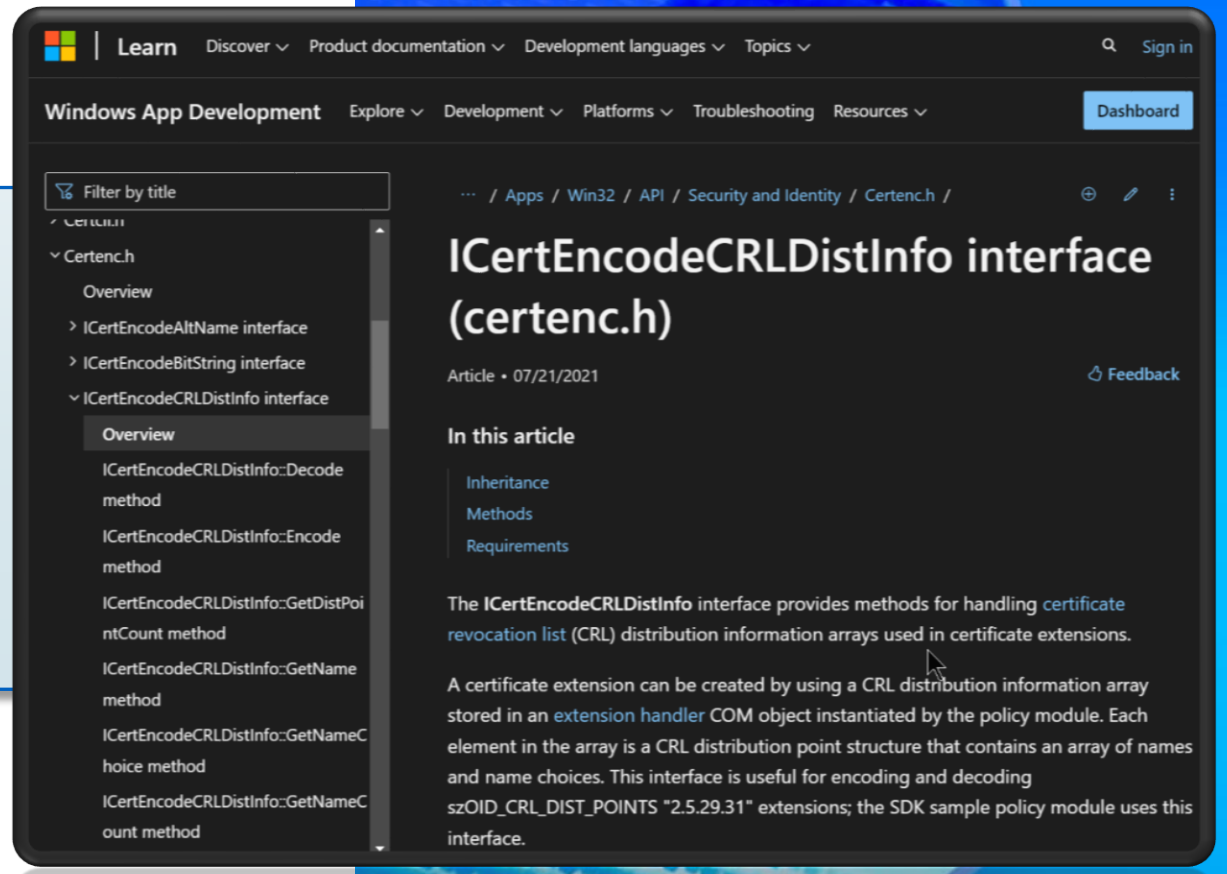
- `szOID_CRL_DIST_POINTS = "2.5.29.31"`
- `Certificate file`
-
-





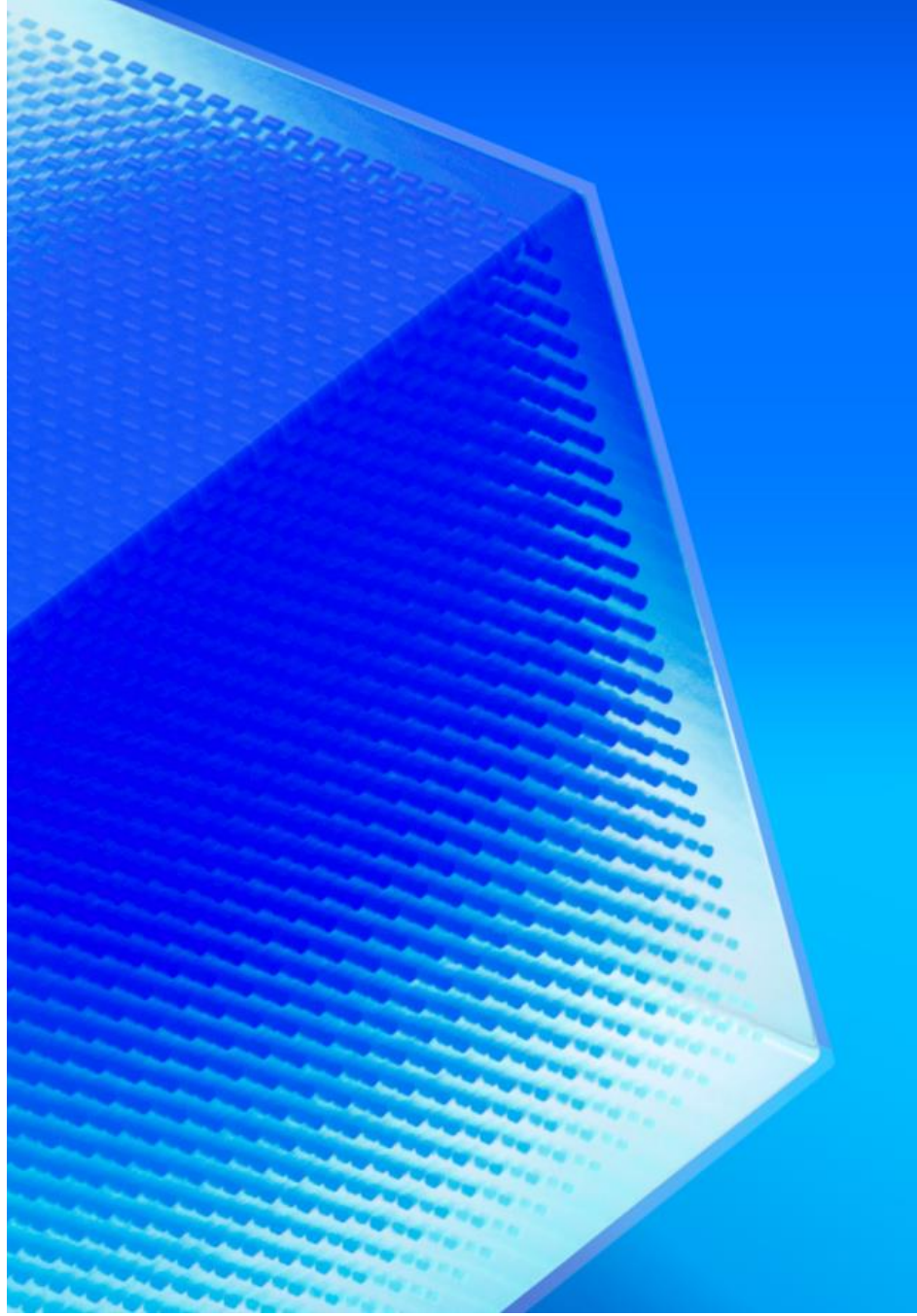
Vulnerability Options

- `szOID_CRL_DIST_POINTS = "2.5.29.31"`
- `Certificate file`
- `COM Interface`
-



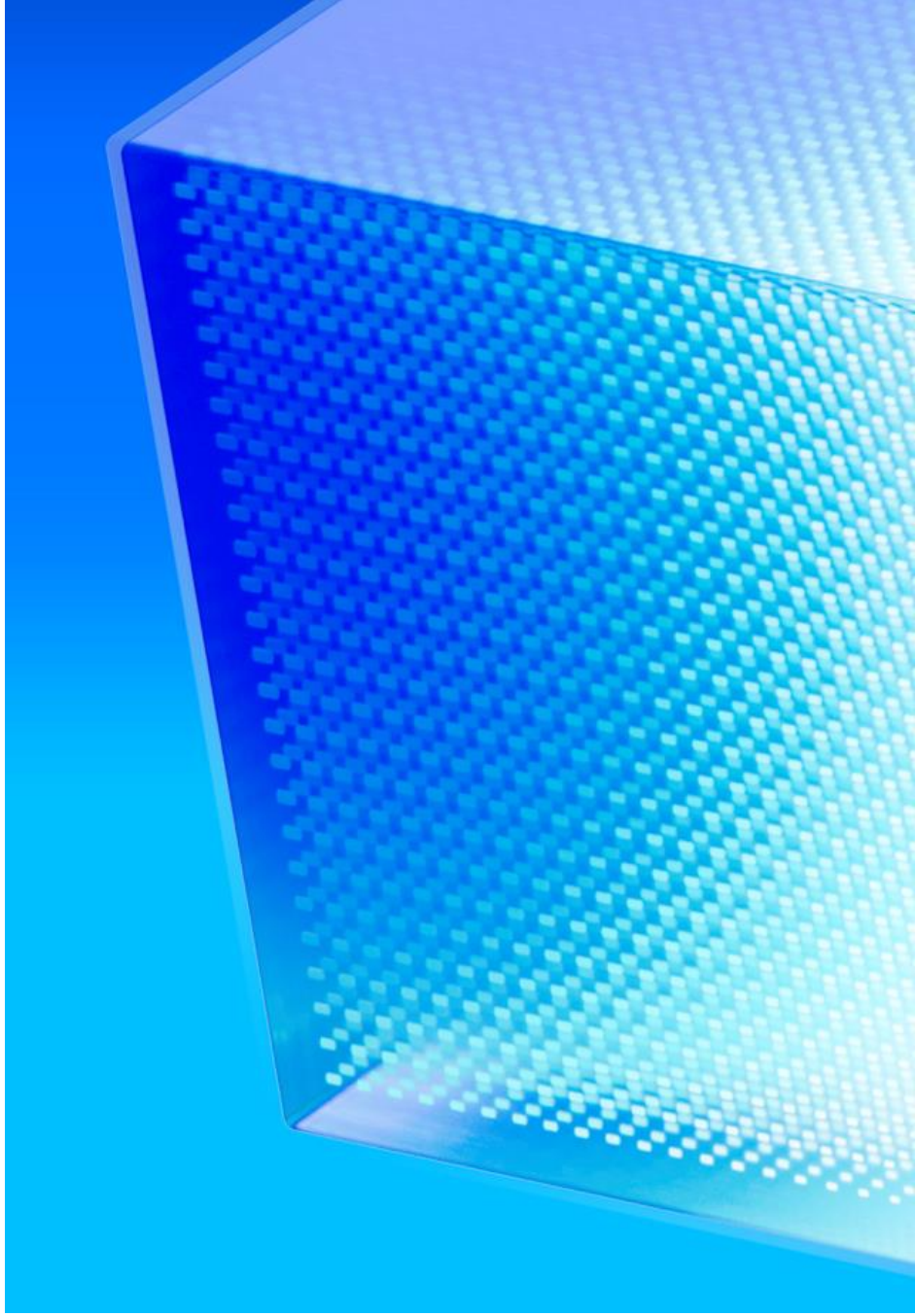
Vulnerability Options

- szOID_CRL_DIST_POINTS = “2.5.29.31”
- Certificate file
- COM Interface
- SSL/TLS Network Connections



Reaching Via SSL

- Certificate in SSL Handshake (client or server*)
- SSL Protocol message limit
  - 16MB
- Certificate Chain of Trust
- Retrieval of Missing Certificates
- Extensions: AIA, OCSP, CRL



SSL Object Retrieval

C++ Copy

```
BOOL CryptRetrieveObjectByUrlW(
  [in]           LPCWSTR                 pszUrl,
  [in]           LPCSTR                  pszObjectOid,
  [in]           DWORD                   dwRetrievalFlags,
  [in]           DWORD                   dwTimeout,
  [out]          LPVOID                  *ppvObject,
  [in]           HCRYPTASYNC             hAsyncRetrieve,
  [in, optional] PCRYPT_CREDENTIALS      pCredentials,
  [in, optional] LPVOID                  pvVerify,
  [in]           PCRYPT_RETRIEVE_AUX_INFO pAuxInfo
);
```

Parameters

`[in] pszUrl`

The address of a PKI object to be retrieved. The following schemes are supported:

- ldap (Lightweight Directory Access Protocol)
- http
- https (certificate revocation list (CRL) or online certificate status protocol (OCSP) retrievals only)
- file

C++ Copy

```
typedef struct _CRYPT_RETRIEVE_AUX_INFO {
  DWORD                                cbSize;
  FILETIME                             *pLastSyncTime;
  DWORD                                dwMaxUrlRetrievalByteCount;
  PCRYPTNET_URL_CACHE_PRE_FETCH_INFO   pPreFetchInfo;
  PCRYPTNET_URL_CACHE_FLUSH_INFO       pFlushInfo;
  PCRYPTNET_URL_CACHE_RESPONSE_INFO    *ppResponseInfo;
  LPWSTR                               pwszCacheFileNamePrefix;
  LPFILETIME                           pftCacheResync;
  BOOL                                 fProxyCacheRetrieval;
  DWORD                                dwHttpStatusCode;
  LPWSTR                               *ppwszErrorResponseHeaders;
  PCRYPT_DATA_BLOB                     *ppErrorContentBlob;
} CRYPT_RETRIEVE_AUX_INFO, *PCRYPT_RETRIEVE_AUX_INFO;
```

Members

`dwMaxUrlRetrievalByteCount`

A value that specifies a limit to the number of bytes retrieved. A value of zero or less specifies no limit.

CA Issuers

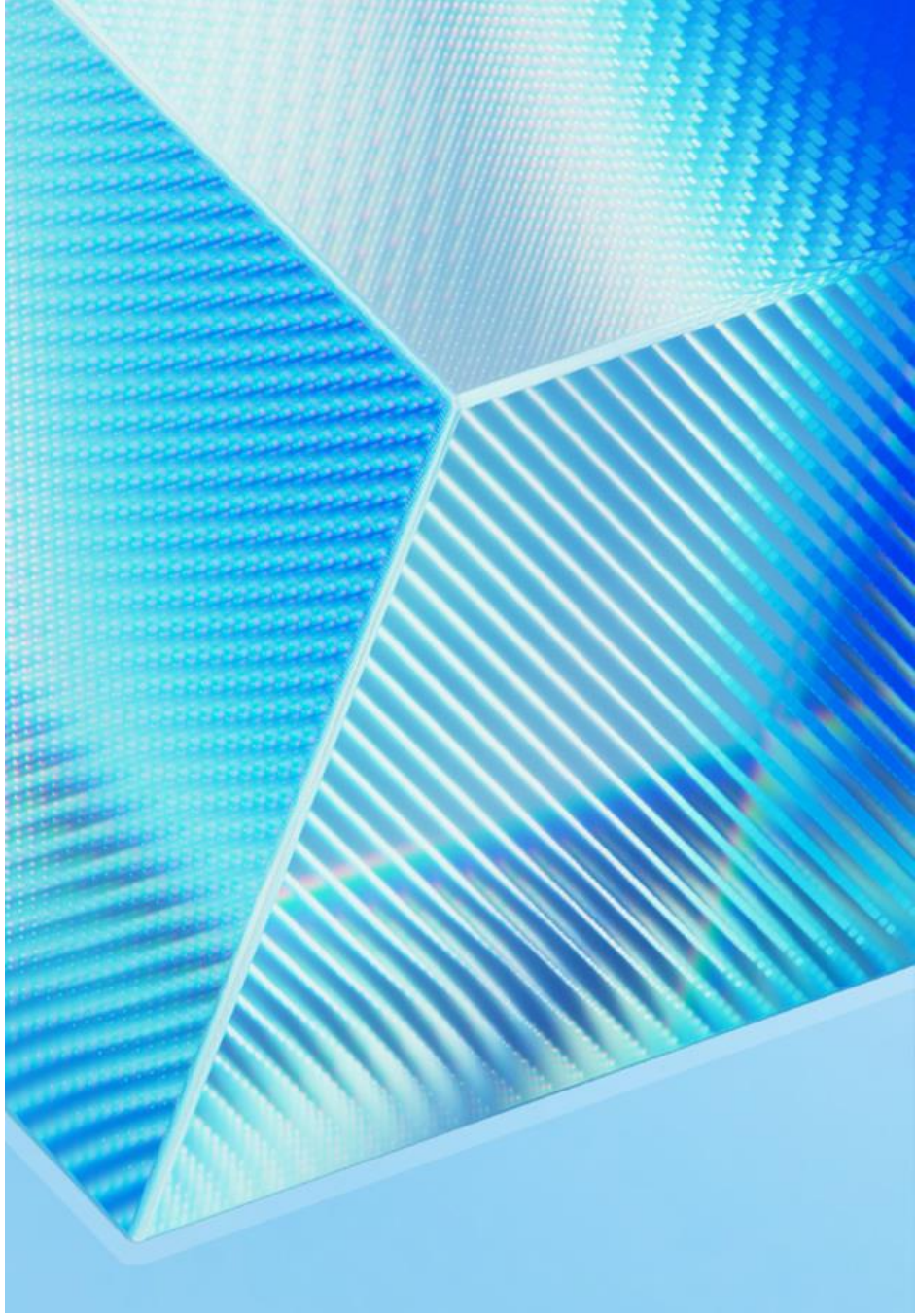
```
X509v3 extensions:
    X509v3 Authority Key Identifier:
        9D:6E:82:D9:A6:69:4D:B2:CA:D1:8D:21:89:41:82:92:02:C2:C9:D4
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
    X509v3 Subject Alternative Name:
        DNS:DirectChild
    Authority Information Access:
        CA Issuers - URI:http://192.168.37.1:8080/intermediate.cer
```

```
CCertChainEngine::GetIssuerUrlStore()
{
    pAuxInfo->dwMaxUrlRetrievalByteCount = 100000;
}
```

FIELD EFFECT

UserNotice Fuzzing

```
                WinAFL 1.17 based on AFL 2.43b (FuzzX509.exe)

+- process timing ------------------------------------+- overall results ----+
|        run time : 1 days, 4 hrs, 12 min, 24 sec     |  cycles done : 40     |
|   last new path : 0 days, 3 hrs, 37 min, 39 sec     |  total paths : 1833   |
| last uniq crash : none seen yet                     | uniq crashes : 0      |
|  last uniq hang : none seen yet                     |   uniq hangs : 0      |
+- cycle progress --------------------+- map coverage -+-----------------------+
|  now processing : 1370* (74.74%)    |    map density : 0.30% / 9.78%         |
| paths timed out : 0 (0.00%)         | count coverage : 1.99 bits/tuple       |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : arith 8\8             | favored paths : 725 (39.55%)           |
| stage execs : 73.3k/611k (11.99%)   |  new edges on : 997 (54.39%)           |
| total execs : 99.7M                 | total crashes : 0 (0 unique)           |
|  exec speed : 410.3/sec             |  total tmouts : 0 (0 unique)           |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 269/8.78M, 100/8.78M, 58/8.78M        |    levels : 22         |
|  byte flips : 3/1.10M, 5/248k, 4/279k               |   pending : 123        |
| arithmetics : 331/12.5M, 7/3.09M, 5/488k            |  pend fav : 0          |
|  known ints : 71/1.05M, 63/7.77M, 42/10.7M          | own finds : 1832       |
|  dictionary : 0/0, 0/0, 34/18.1M                    |  imported : n/a        |
|       havoc : 840/17.9M, 0/0                        | stability : 89.15%     |
|        trim : 26.07%/82.3k, 78.59%                  +------------------------+
+-----------------------------------------------------+   [cpu000001:   4%]
```



FIELD EFFECT

OCSP Retrieval

```
X509v3 extensions:
    X509v3 Authority Key Identifier:
        39:31:A6:FC:DC:78:2B:B6:59:59:93:F1:BE:50:AB:EC:F5:1E:08:9F
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
    Authority Information Access:
        OCSP - URI:http://192.168.37.1:8080/ocsp
```

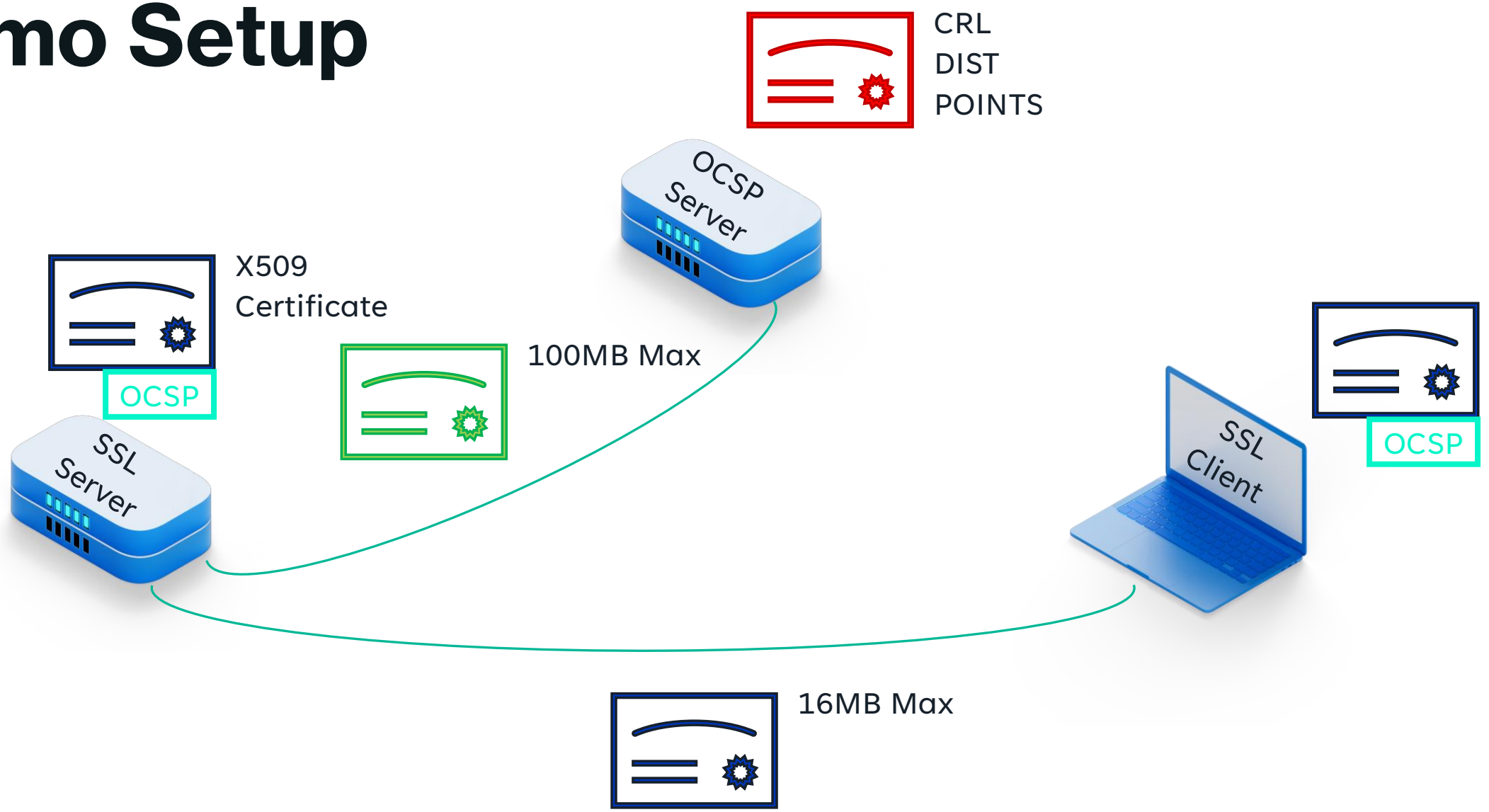
```
pAuxInfo->dwMaxUrlRetrievalByteCount = 100MB;
```

```
3.2  Signed Response Acceptance Requirements

   Prior to accepting a signed response as valid, OCSP clients SHALL
   confirm that:

   1. The certificate identified in a received response corresponds to
   that which was identified in the corresponding request;

   2. The signature on the response is valid;

   3. The identity of the signer matches the intended recipient of the
   request.

   4. The signer is currently authorized to sign the response.

   5. The time at which the status being indicated is known to be
   correct (thisUpdate) is sufficiently recent.

   6. When available, the time at or before which newer information will
   be available about the status of the certificate (nextUpdate) is
   greater than the current time.
```

Dead End?

- Compromised Intermediate CA (e.g. Comodo)
- Firewalls/Security Appliance Inspecting SSL
- Independently Manage CAs

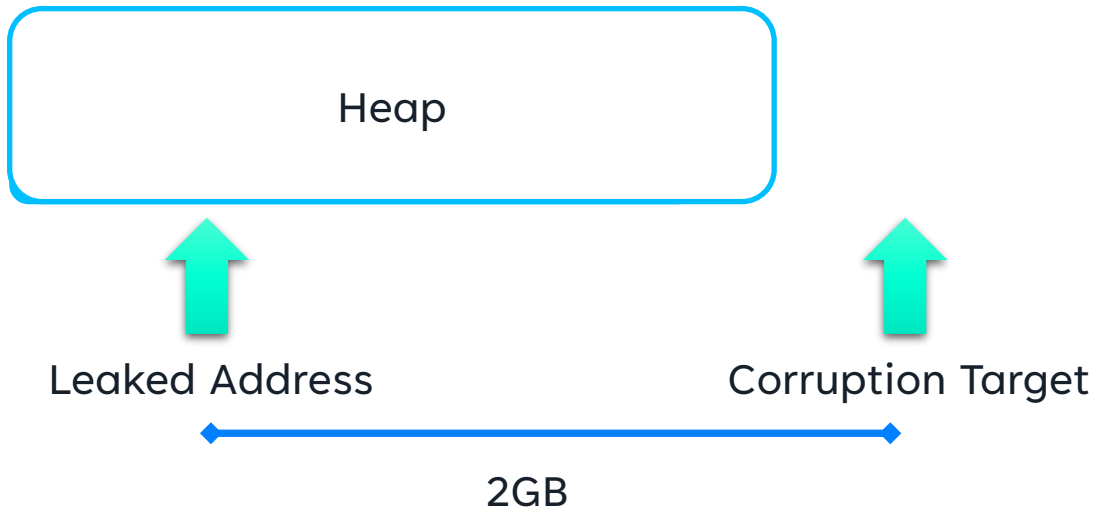
Demo Setup


CRL
DIST
POINTS
OCSP
Server
X509
Certificate
OCSP
100MB Max
SSL
Server
SSL
Client
OCSP
16MB Max

CVE-2024-29050

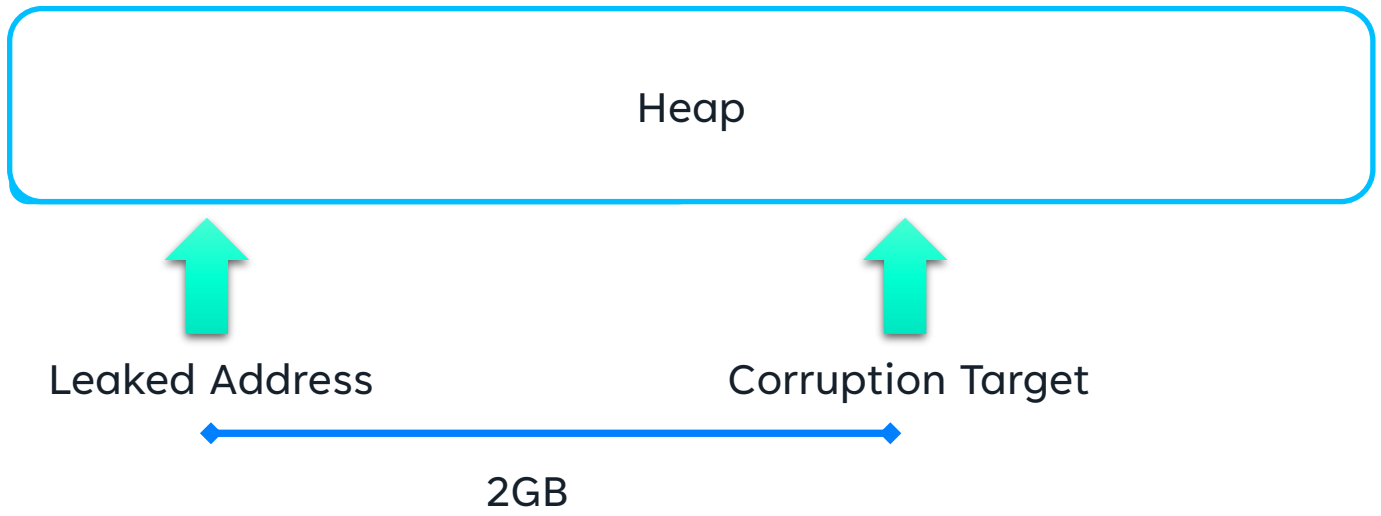
Exploitable?

- Primitive is write of controlled data 2GB from allocation



Exploitable?

- Primitive is write of controlled data 2GB from allocation
- Misses will likely be swallowed by exception handler
- Opportunity for info leaks
- Sensitive data in LSASS, don't need RCE to win



FIELD EFFECT

Disclosure Timelines

CVE-2024-30020 - Time Stamp Response

- 2023-12 – Found and reported to MSRC
- 2024-02 – Vulnerability confirmed by MSRC
- 2024-05 – Patch released

CVE-2024-29050 - CRL Distribution Points

- 2023-09 - Reported to MSRC by VictorV with Kunlun Lab
- 2023-12 – Found and reported by me to MSRC
- 2024-02 – Vulnerability confirmed by MSRC
- 2024-04 – Patch released

FIELD EFFECT

Thank you!

- eegsgard@fieldeffect.com
- @hexnomad@infosec.exchange