Hacking Exchange From the OutsideIn

whoami /all

- Senior Research Consultant – Atredis Partners
- 6 years Industry Experience
- Hobbies:
  - Windows Malware Techniques
  - Windows Exploitation
  - Powerlifting
  - Heavily Edited Photography
  - Mafia (board game)
- Twitter / GitHub: @aahmad097

Overview

- Fuzzing Target Selection
- Fuzzer Selection
- Harness “Development”
- Fuzzing Optimization
- Results
- Target Selection
- Target Environment Setup
- Reporting Process

Fuzzing Target Selection

Oracle Outside In Technology

Set of SDKs that parse files

Developed using C++

Oracle OutsideIn Technology

- Used for content
  - **Extraction**
  - Normalization
  - Scrubbing
  - Conversion
  - Viewing
- 600 Supported file formats
  - Including: **PDF**, **Excel**, Word, **HTML**, XML

Fuzzer Types

- Dumb Fuzzers
- Grammar-based Fuzzers
- Mutation-based Fuzzers
- Coverage-guided Fuzzers
- Protocol Fuzzers
- Etc.

Fuzzer Types

- Dumb Fuzzers
  - Pros
    - It’s simple?
    - No knowledge of target required
  - Cons
    - No understanding of target
    - Not very useful for complex targets

Fuzzer Types

- Grammar-based Fuzzers
  - Pros
    - Deep understanding of target
    - Generates correct syntax
  - Cons
    - Dependent on fuzzing grammar
    - Labor Intensive

Fuzzer Types

- Mutation-based Fuzzers
  - Pros
    - Ease of setup and use
    - Large number of testcases can be generated quickly
  - Cons
    - Limited by the corpus provided
    - Not as effective for complex formats

Fuzzer Types

- Coverage Guided Fuzzers
  - Pros
    - High code coverage
    - Great for complex targets
  - Cons
    - Can be resource intensive for instrumentation
    - Initial setup can be difficult

Fuzzer Types

- Protocol Fuzzers
  - Pros
    - Targeted protocol testing
    - Discovering critical vulnerabilities
  - Cons
    - Complex configuration
    - Resource intensive
    - Very specific targets

Fuzzer Selection

- Linux
  - **American Fuzzy Lop (AFL) + DynInst** (selected for familiarity)
    - Mutation / Coverage based fuzzer
- Windows
  - **Jackalope + TinyInst** (selected for ease of setup / use on windows)
    - Mutation / Coverage based fuzzer

American Fuzzy Lop - https://github.com/google/AFL
DynInst - https://github.com/talos-vulndev/afl-dyninst
Jackalope - https://github.com/googleprojectzero/Jackalope
TinyInst - https://github.com/googleprojectzero/TinyInst

Linux Setup

- Clone AFL (v2.57b)
- Clone afl-dyninst (v9.3.1)
- Compile Applications

Additional Setup (Linux)

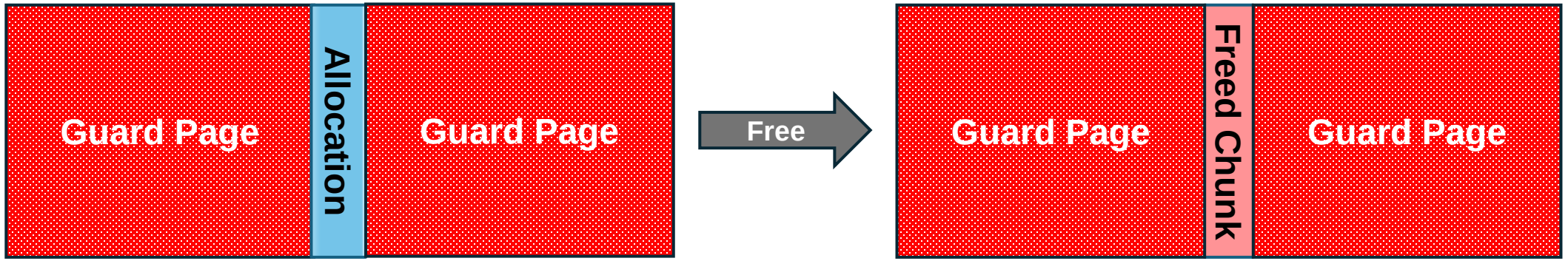
- Static Instrumentation
  - Instrumenting target libraries
  - Min basic block (8 bytes)
    - Due to stability

Windows Setup

- Install Visual Studio and CMake
- Clone Jackalope
- Clone TinyInst and Dependencies
- Compile and Build with CMake

Additional Setup (Windows)

- Windows SDK (gflags)
  - Full Page Heap
    - Guard Pages
    - Fills Uninitialized Memory (c0c0c0c0 pattern)
    - Marks freed allocations as no access
    - Usage: gflags.exe /i <program_name> +hpa



Harness “Development”

- Lightweight program that calls target functions
- Handles environment setup
- Requires some understanding of target functions and input

Memoryio

- Example Usage of OIT CA
- Reads file and passes bytes to OIT I/O routines
- Used DA modules
- Usage:
  - memoryio.exe <input_file>

Fuzzer and Harness Optimization

- Harness Modifications
  - Stripping debugging statements (printf and the like)
  - Implementing Shared Memory
- Fuzzer Modification (technically instrumentation library)
  - Persistent fuzzing bug - stack variable overwriting lcContext value

Implementing Shared Memory

- HANDLE OpenFileMappingA(
  [in] DWORD dwDesiredAccess,
  [in] BOOL bInheritHandle,
  [in] LPCSTR lpName);

- LPVOID MapViewOfFile(
  [in] HANDLE hFileMappingObject,
  [in] DWORD dwDesiredAccess,
  [in] DWORD dwFileOffsetHigh,
  [in] DWORD dwFileOffsetLow,
  [in] SIZE_T dwNumberOfBytesToMap);

```
map_file = OpenFileMappingA(
    FILE_MAP_ALL_ACCESS,   // read/write access
    FALSE,                 // do not inherit
    lpPath);            // name of mapping object


shm_data = (unsigned char*)MapViewOfFile(
    map_file, // handle to map object
    FILE_MAP_ALL_ACCESS,  // read/write permission
    0, // high-order offset of view
    0, // low-order offset of view
    NULL); // # bytes to map (NULL = Full File)
```

Fuzzer Execution


```
C:\Users\AliSAhmad\source\repos\Jackalope\build\Release\fuzzer.exe^
    -in "IN\pdf"^
    -out "Out\pdf"^
    -t 5000^
    -nthreads 20^
    -delivery shmem^
    -nargs 2^
    -instrument_module vspdf.dll^
    -target_module memoryio_sharedmem.exe^
    -target_offset 0x2900^
    -dump_coverage^
    -persist^
    -loop^
    -iterations 5000^
    -dict "C:\Users\AliSAhmad\source\repos\AFLplusplus\dictionaries\pdf.dict"^
    -- "C:\Users\AliSAhmad\Documents\Research\Oracle Outside In - Access Manager\Exchange
\TE_v.8.5.3.0\memoryio_sharedmem.exe" @@
```

Helpful Debugger Breakpoints

- Break on module load
  - sxe ld vshtml.dll
- Break on specific hit
  - bu <location> "r $t0 = @$t0 + 1; .if (@$t0 < 0x37d) { gc; } .else { .echo 'Value reached or exceeded 0x37d'; }"
- Logging Allocations
  - bu ntdll!rtlallocateheap "r $t0 = @rcx; r $t1 = @rdx; r $t2 = @r8; g“
  - bu ntdll+??? ".printf \" %p = RtlAllocateHeap(%p, %d, %d)\\n\", @rax, $t0, $t1, $t2; gc“
  - ??? – Offset to RtlAllocateHeap return

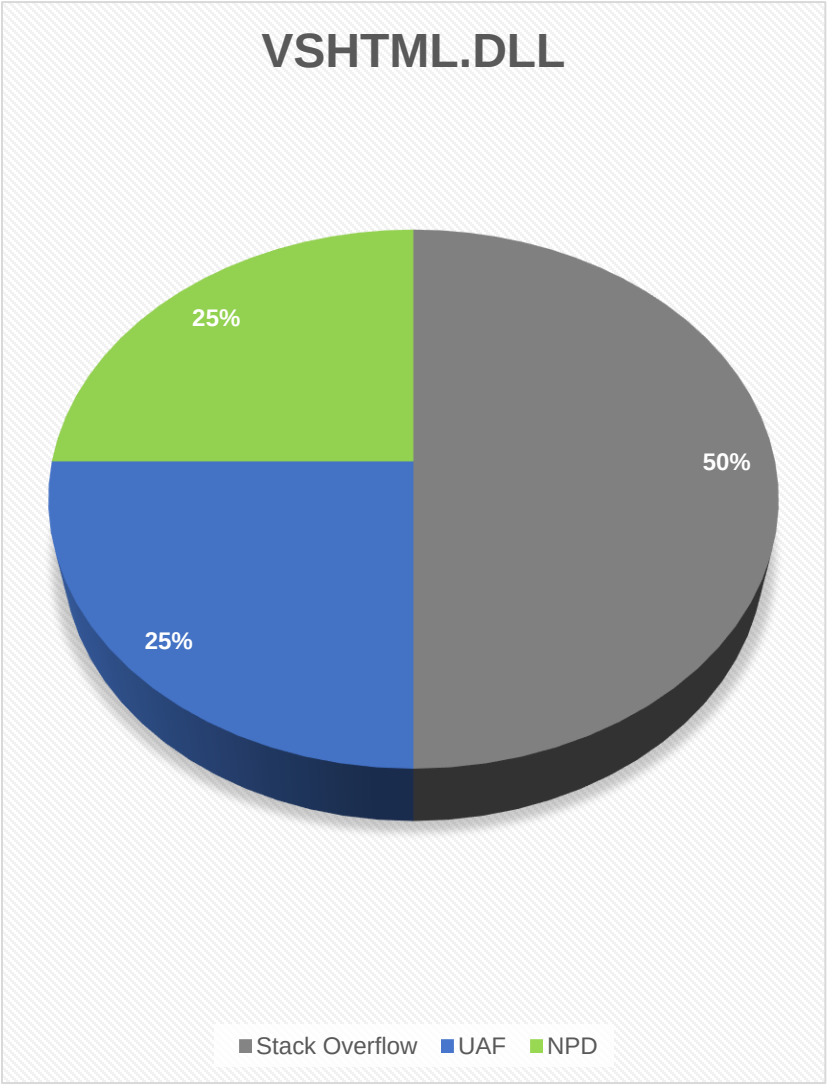
Fuzzing Results

- Libraries Fuzzed
  - VSXL5.DLL – responsible for XL5 format parsing
  - VSPDF.DLL – responsible for PDF format parsing
  - VSHTML.DLL – responsible for HTML format parsing

- Unintentional Results
  - SCCUT.DLL - Utility library used in OIT

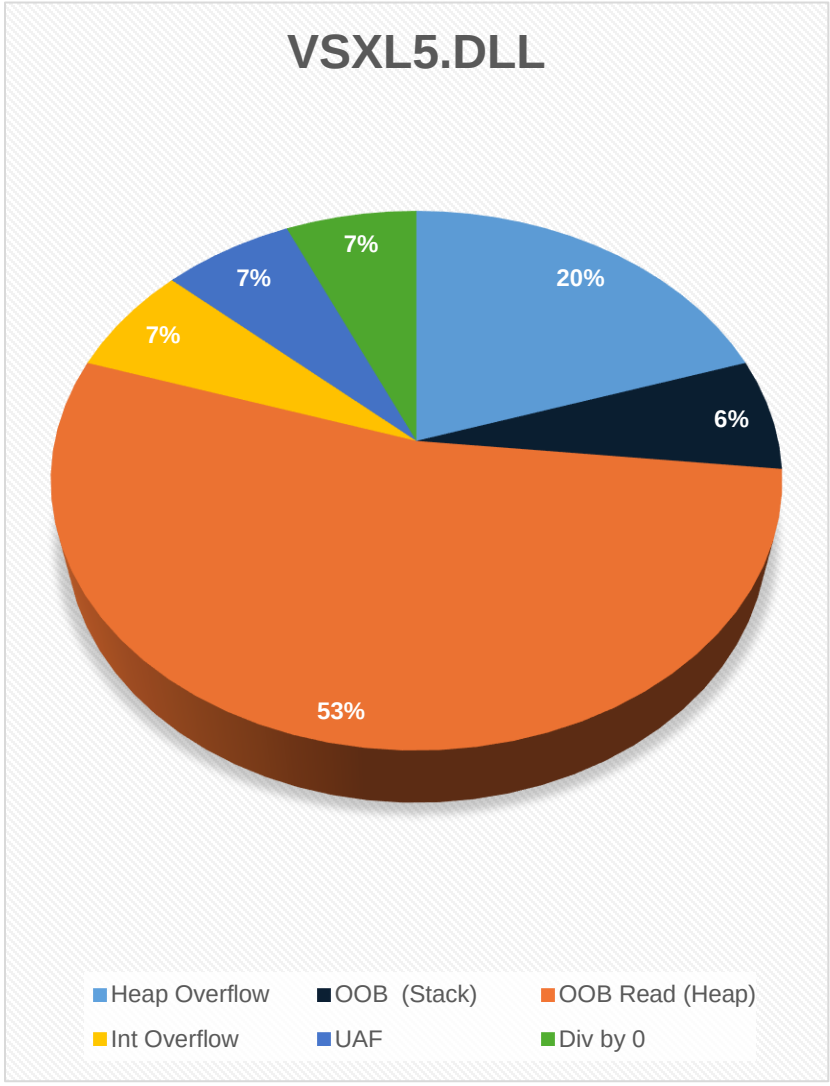
VSHTML.DLL

- Null Pointer Dereference: 1
- Stack Overflow: 2
- Use After Free : 1



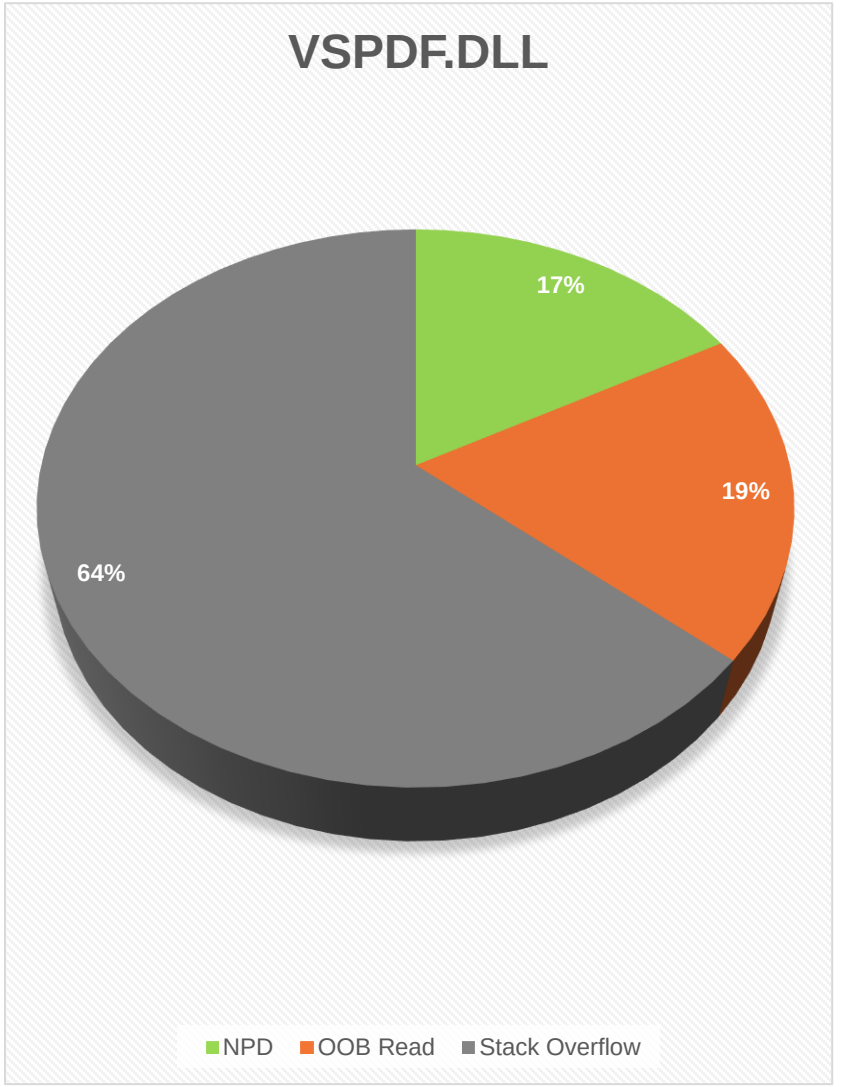
VSXL5.DLL

- Heap Overflow: 3
- Heap Out of Bounds Read: 9
- Out of Bound Stack: 1
- Integer Overflow: 1
- Use After Free: 1
- Divide by 0: 1



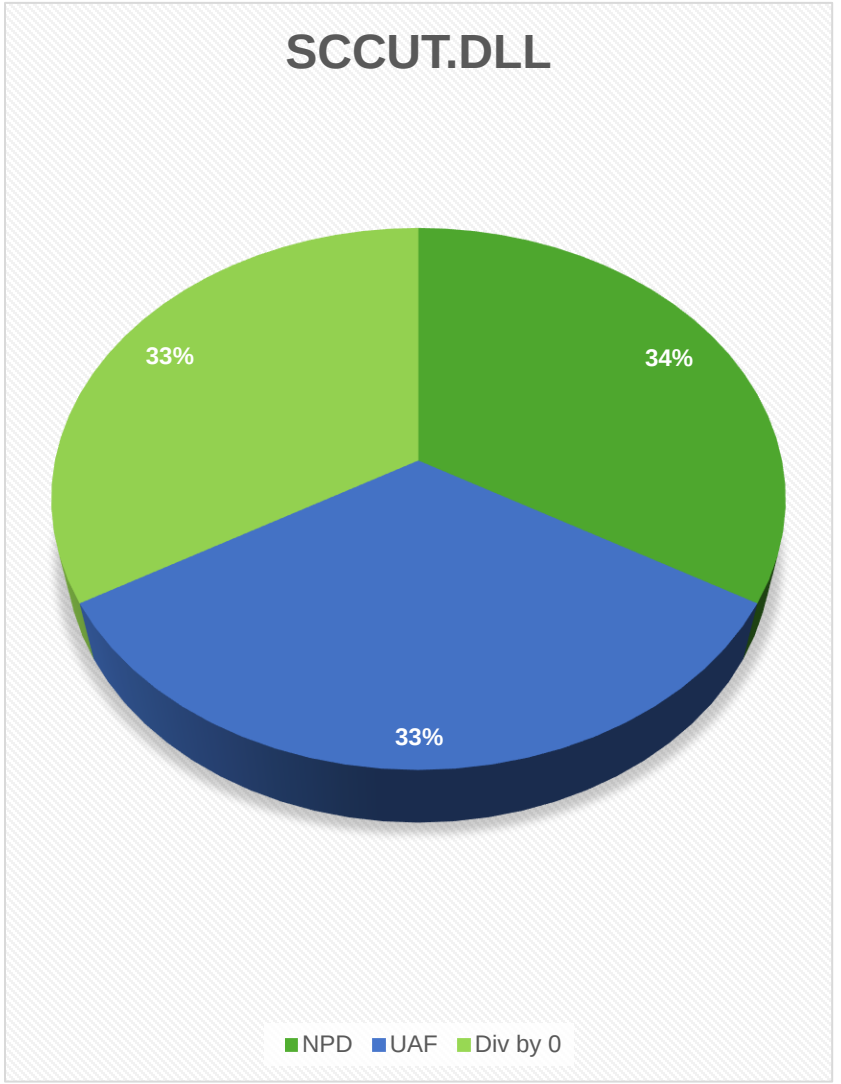
VSPDF.DLL

- Stack Overflow: 30
- Null Pointer Dereference: 8
- Heap Out of Bounds Read: 9

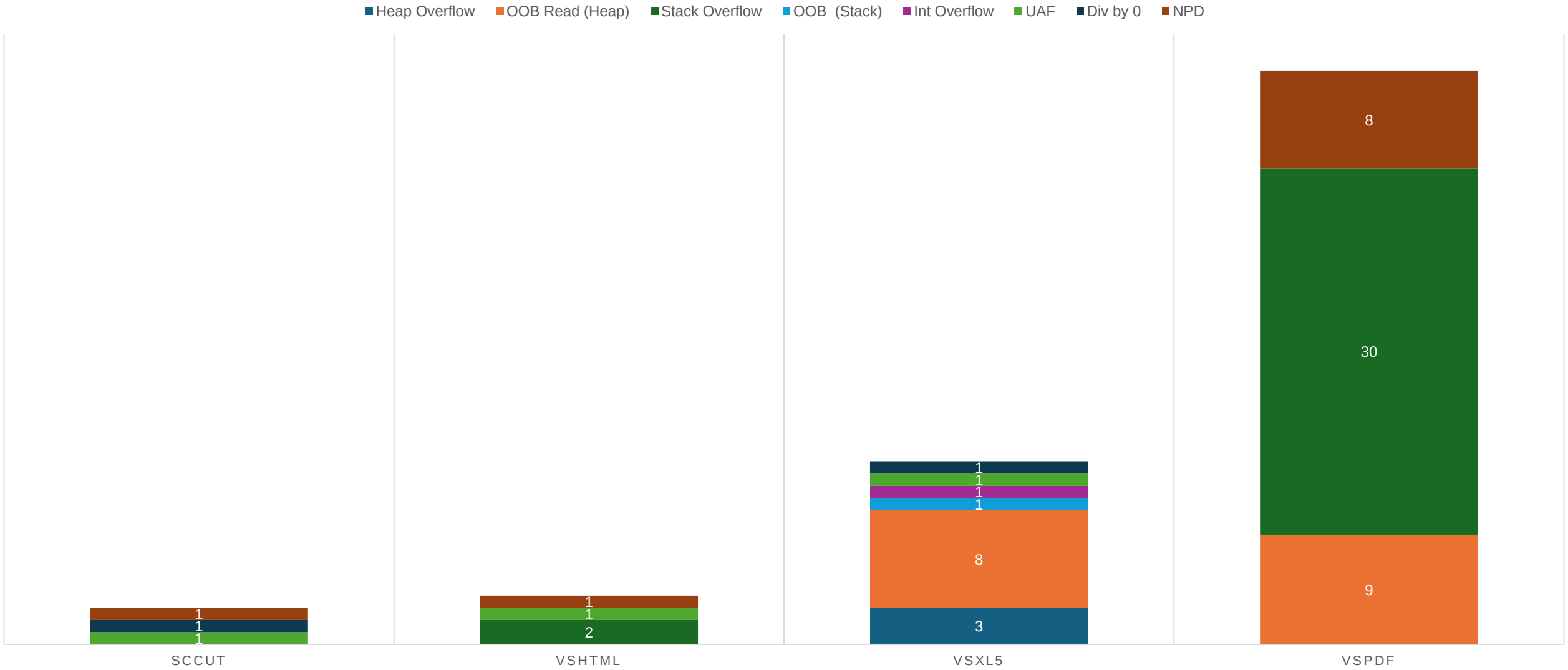


SCCUT.DLL

- Use After Free: 1
- Null Pointer Dereference: 1
- Divide By 0: 1



Big Picture


Heap Overflow
OOB Read (Heap)
Stack Overflow
OOB (Stack)
Int Overflow
UAF
Div by 0
NPD
1
1
1
1
1
2
1
1
1
1
8
3
8
30
9
SCCUT
VSHTML
VSXL5
VSPDF

Case Study – UAF VSHTML

Fuzzer Output:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "xhtml1-frameset.dtd">
<a�se href="about:blank" target="Frame1"/>
<frameset cols="20, 80">
<frameset rows="100, 200">
<frame marginheight="10" marginwidth="5" noresize="noresize" name="Frame1" frameborder="1" scrolling="yes" src="right.png" />
</frameset>
</html>
```

Minified Sample:

```
<!DOCTYPE html><a><frameset>
```

Case Study – UAF VSHTML Demo

Case Study – UAF VSHTML


```
node_wrapper *__fastcall disconnectAndOptionallyDeleteNode(node_wrapper *HtmlNode, BOOL bDelete)
{
  node_wrapper *parentNode; // rax
  node_wrapper *parent; // rcx
  node_wrapper *leftSibling; // rcx
  node_wrapper *rightSibling; // rcx

  HtmlNode->vtable = (__int64)&oit::LeftChildRightSibTreeNode<HtmlGenericNodeData *>::`vftable';
  parentNode = (node_wrapper *)HtmlNode->parent;
  if ( parentNode )
  {
    --parentNode->childcount;                   // decrement childcount of parent node
    parent = (node_wrapper *)HtmlNode->parent;
    if ( (node_wrapper *)parent->leftmost_child == HtmlNode )// if node is left child
      parent->leftmost_child = HtmlNode->right_sibling;// remove parent left most child relationship
    HtmlNode->parent = 0i64;                    // remove node parent pointer
  }
  leftSibling = (node_wrapper *)HtmlNode->left_sibling;
  if ( leftSibling )
    leftSibling->right_sibling = HtmlNode->right_sibling;// remove left sibling relationship
  rightSibling = (node_wrapper *)HtmlNode->right_sibling;
  if ( rightSibling )
    rightSibling->left_sibling = HtmlNode->left_sibling;// remove right sibling relationship
  HtmlNode->left_sibling = 0i64;                // remove left sibling pointer
  HtmlNode->right_sibling = 0i64;               // remove right sibling pointer
  if ( bDelete )
    operator delete(HtmlNode);                  // free node
  return HtmlNode;
}
```

Logic exists to remove parent's left child but not right

Case Study – UAF VSHTML


```
if ( NodeToBeFreed )
{
  initialParentNode = *(node_wrapper **)(vwstream->initialParentNode + 0x20);
  if ( NodeToBeFreed->parent )
  {
    _10_HTMLObjectNode = (void (__fastcall ***)(_QWORD, __int64))NodeToBeFreed->_10_HTMLObjectNode;
    --LODWORD(initialParentNode->_10_HTMLObjectNode);
    (*(void (__fastcall **)(node_wrapper *, __int64))NodeToBeFreed->vtable)(NodeToBeFreed, 1i64); // Disconnect Node and Free
  }
  ... snip ...
  if ( _10_HTMLObjectNode )
    (**_10_HTMLObjectNode)(_10_HTMLObjectNode, 1i64); //HtmlTagDestructor
}
CreateHTMLNode(intFlag, 1, vwstream, 1); // Use
... snip ...
v8 = 0;
}
goto LABEL_37;
}
```

Case Study – UAF VSHTML Patch


```
node_wrapper *__fastcall removeNodesRelationships(node_wrapper *HtmlNode)
{
  node_wrapper *tmpNode; // rax
  node_wrapper *parent; // rdx
  node_wrapper *left_sibling; // rdx
  node_wrapper *right_sibling; // rdx

  tmpNode = (node_wrapper *)HtmlNode->parent;
  if ( tmpNode )
  {
    --tmpNode->childcount;                      // decrement parent child count
    parent = (node_wrapper *)HtmlNode->parent;
    if ( (node_wrapper *)parent->leftmost_child == HtmlNode )// if node is left child
    {
      tmpNode = (node_wrapper *)HtmlNode->right_sibling;
      parent->leftmost_child = (__int64)tmpNode;// set leftmost child pointer to node right sibling
      parent = (node_wrapper *)HtmlNode->parent;
    }
    if ( (node_wrapper *)parent->rightmost_child == HtmlNode )// if node is right child
    {
      tmpNode = (node_wrapper *)HtmlNode->left_sibling;
      parent->rightmost_child = (__int64)tmpNode;// set rightmost child to left sibling
    }
    HtmlNode->parent = 0i64;                    // remove node->parent relationship
  }
  left_sibling = (node_wrapper *)HtmlNode->left_sibling;
  if ( left_sibling )                           // if has left sibling
  {
    tmpNode = (node_wrapper *)HtmlNode->right_sibling;
    left_sibling->right_sibling = (__int64)tmpNode;// set left node's right sibling to node's right sibling
  }
  right_sibling = (node_wrapper *)HtmlNode->right_sibling;
  if ( right_sibling )                          // if has right sibling
  {
    tmpNode = (node_wrapper *)HtmlNode->left_sibling;
    right_sibling->left_sibling = (__int64)tmpNode;// set right node's left sibling to node's left sibling
  }
  HtmlNode->right_sibling = 0i64;               // remove node right sibling relationship
  HtmlNode->left_sibling = 0i64;                // remove node left sibling relationship
  return tmpNode;
}
```

Case Study – UAF VSHTML Patch


```
    if ( NodeToBeFreed )
    {
      HtmlTagObj = (void (__fastcall ***)(_QWORD, __int64))NodeToBeFreed->_10_HTMLObjectNode;
      if ( NodeToBeFreed->parent )
      {
        --*(_DWORD *)(*(_QWORD *)(*(_QWORD *)&vwstream->gapEC8[40] + 32i64) + 16i64); // decrement inital node child count
        FreeNodeChildren(NodeToBeFreed);
      }
      (*(void (__fastcall **)(node_wrapper *, __int64))NodeToBeFreed->vtable)(NodeToBeFreed, 1i64); // disconnect and free node
      if ( HtmlTagObj )
        (**HtmlTagObj)(HtmlTagObj, 1i64); // HtmlTagObj Destructor
    }
      ... snip ...
  }
  ... snip ...
  return v9;
}
```

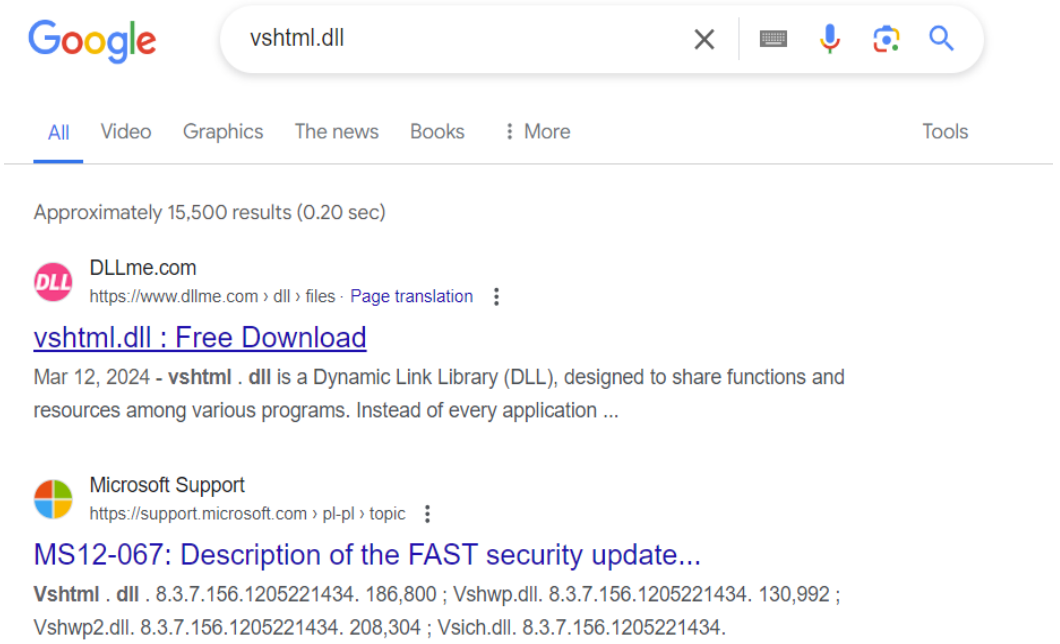
Case Study – UAF VSHTML Patch


```
__int64 __fastcall FreeChildren(node_wrapper *Node)
{
  ... snip ...

  if ( Node->childcount == 1 )
  {
    result = FreeChildren((node_wrapper *)Node->leftmost_child);
    leftmost_child = (node_wrapper *)Node->leftmost_child;
    v4 = (__int64 (__fastcall ***)(_QWORD, __int64))leftmost_child->HtmlTagObj;
    if ( leftmost_child )
      result = (*(__int64 (__fastcall **)(node_wrapper *, __int64))leftmost_child->vtable)(leftmost_child, 1i64);
    if ( v4 )
    {
      result = (**v4)(v4, 1i64);
      Node->rightmost_child = 0i64;
      return result;
    }
  }
  else
  {
    if ( Node->childcount <= 1u )
      return result;
    result = FreeChildren((node_wrapper *)Node->leftmost_child);
    leftmost_child2 = (__int64 (__fastcall ***)(_QWORD, __int64))Node->leftmost_child;
    if ( leftmost_child2 )
      result = (**leftmost_child2)(leftmost_child2, 1i64);
    rightmost_child = (node_wrapper *)Node->rightmost_child;
    while ( rightmost_child )
    {
      tmpNode = rightmost_child;
      rightmost_child = (node_wrapper *)rightmost_child->left_sibling;
      FreeChildren(tmpNode);
      HtmlTagObj = (__int64 (__fastcall ***)(_QWORD, __int64))tmpNode->HtmlTagObj;
      result = (*(__int64 (__fastcall **)(node_wrapper *, __int64))tmpNode->vtable)(tmpNode, 1i64);
      if ( HtmlTagObj )
        result = (**HtmlTagObj)(HtmlTagObj, 1i64);
    }
  }
  Node->rightmost_child = 0i64;
  return result;
}
```

Target Selection

- Google Search for VSHTML.DLL
- Products
  - **Microsoft Exchange**
  - Oracle SQL Server
  - Oracle WebCenter Content Server



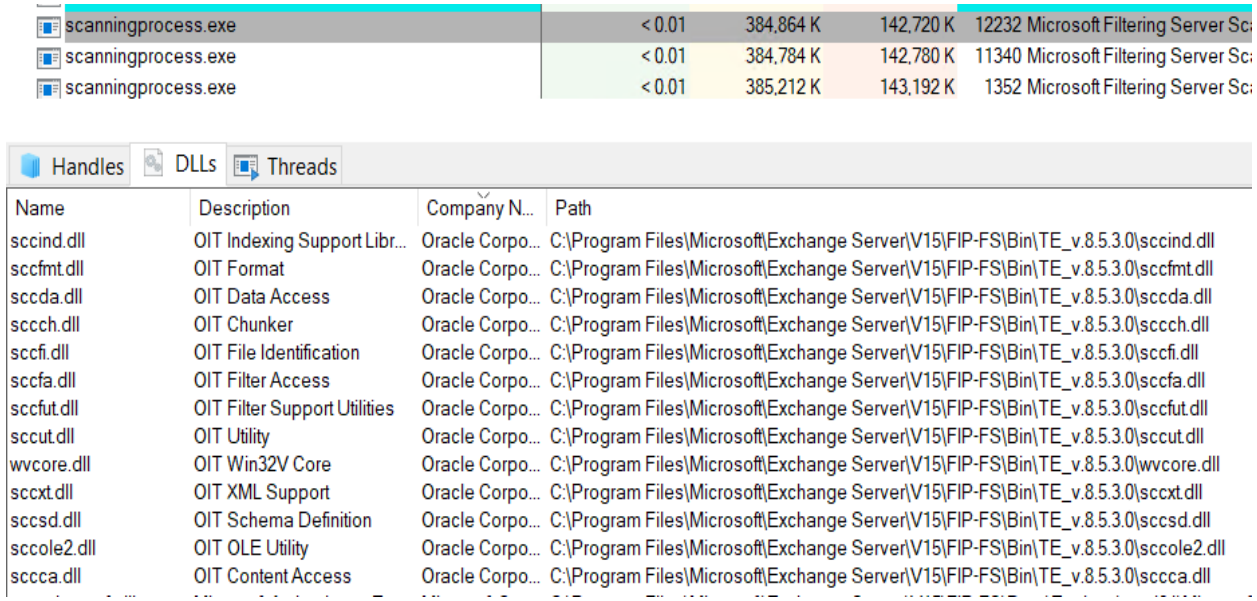
Target Environment Setup

- 2 Windows Servers
  - Domain Controller
  - Exchange Server
- Windows SDK on Exchange Server
  - For gflags
- Use static IP addresses
- Remember to snapshot instances

Fast Information Process Filter Service (FIP-FS)

- Uses
  - Data Loss Prevention (DLP)
  - Malware Detection
  - Spam Detection

- Scanningprocess.exe
  - Component utilizing OIT



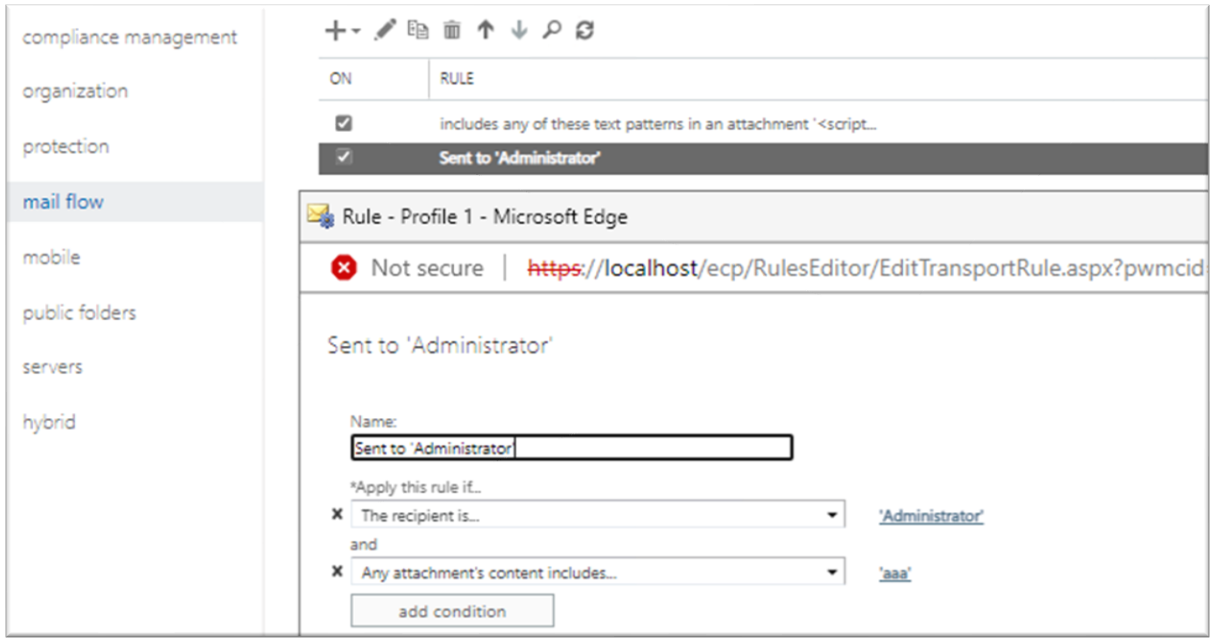
scanningprocess.exe	< 0.01	384,864 K	142,720 K	12232 Microsoft Filtering Server Sc
scanningprocess.exe	< 0.01	384,784 K	142,780 K	11340 Microsoft Filtering Server Sc
scanningprocess.exe	< 0.01	385,212 K	143,192 K	1352 Microsoft Filtering Server Sc

Handles | DLLs | Threads

Name	Description	Company N...	Path
sccind.dll	OIT Indexing Support Libr...	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccind.dll
sccfmt.dll	OIT Format	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccfmt.dll
sccda.dll	OIT Data Access	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccda.dll
sccch.dll	OIT Chunker	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccch.dll
sccfi.dll	OIT File Identification	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccfi.dll
sccfa.dll	OIT Filter Access	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccfa.dll
sccfut.dll	OIT Filter Support Utilities	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccfut.dll
sccut.dll	OIT Utility	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccut.dll
wvcore.dll	OIT Win32V Core	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\wvcore.dll
sccxt.dll	OIT XML Support	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccxt.dll
sccsd.dll	OIT Schema Definition	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccsd.dll
sccole2.dll	OIT OLE Utility	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccole2.dll
sccca.dll	OIT Content Access	Oracle Corpo...	C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Bin\TE_v.8.5.3.0\sccca.dll

Target Environment Configuration

- Add mail flow rules
  - ECP panel on exchange
    - Mail flow
      - Add new attachment rule



Target Environment Configuration

- Modify Configuration
  - C:\Program Files\Microsoft\Exchange Server\V15\FIP-FS\Data\Configuration.xml
  - ScanProcessMemoryMax – 8192
  - ScanProcessMemoryCap – 10240

```
<MemoryUsage>
  <MonitoringMode>Restart</MonitoringMode>
  <ScanProcessMemoryMax>8192</ScanProcessMemoryMax>
  <ScanProcessMemoryCap>10240</ScanProcessMemoryCap>
  <MemoryLeakThreshold>10000</MemoryLeakThreshold>
  <MaxThresholdCrossings>20</MaxThresholdCrossings>
  <SampleRate>1000</SampleRate>
  <SampleSize>120</SampleSize>
</MemoryUsage>
```

Target Environment Configuration

- Enable Full Page Heap
  - gflags.exe /i scanningprocess.exe +hpa

```
C:\Users\administrator.ROOTD>gflags.exe /i scanningprocess.exe +hpa
Current Registry Settings for scanningprocess.exe executable are: 02000000
    hpa - Enable page heap

C:\Users\administrator.ROOTD>
```

Triggering Crash / Catching Trace

- 3 Instances of scanningprocess running
- Send blank email to target
  - Identify vshtml.dll module load
- Attach Debugger
- Send malicious attachment to catch the crash

Reporting Timeline

- 12/13/2023 - Initial Report to MSRC
- 12/20/2023 - MSRC Acknowledgement
- 12/20/2023 - MSRC Query on VSHTML Trigger
- 12/21/2023 - Updated Triggers Sent
- 01/02/2024 - MSRC Acknowledgement
- 01/16/2024 - Oracle Patch (CVE-2024-20930)
- 01/16/2024 - Microsoft Request to Hold of Disclosure
- 03/12/2024 - Microsoft Advisory Decommissioning OIT (ADV24199947)
- 04/15/2024 - Acknowledgement From Microsoft

Acknowledgements

- Atredis Partners:
  - Nick Nam
  - Brandon Perry
  - Jordan Whitehead
- Microsoft:
  - Lisa Olson (MSRC)