black hat ASIA 2024

APRIL 18-19, 2024

BRIEFINGS

A Glimpse Into The Protocol Fuzz Windows RDP Client For Fun And Profit

Yingqi Shi(@Mas0nShi), Mingjia Liu(@cyberestro), Quan Jin(@jq0904)

DBAPPSecurity

About Us

Yingqi Shi
@Mas0nShi

Mingjia Liu
@cyberestro

Quan Jin
@jq0904

Guoxian Zhong
@_p01arisZ

Siyuan Liu
@4nsw3r123

安恒信息
DAS-security 安全中国

卫兵实验室
WEBIN.LAB

Agenda

Motivation

Introduction

Fuzzing

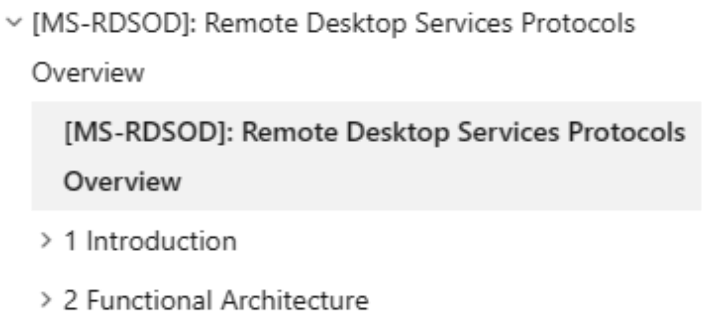
Case Study

Future

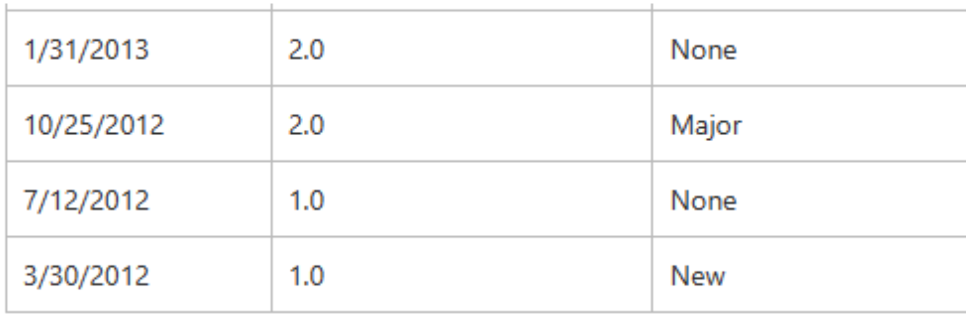
Motivation

Motivation

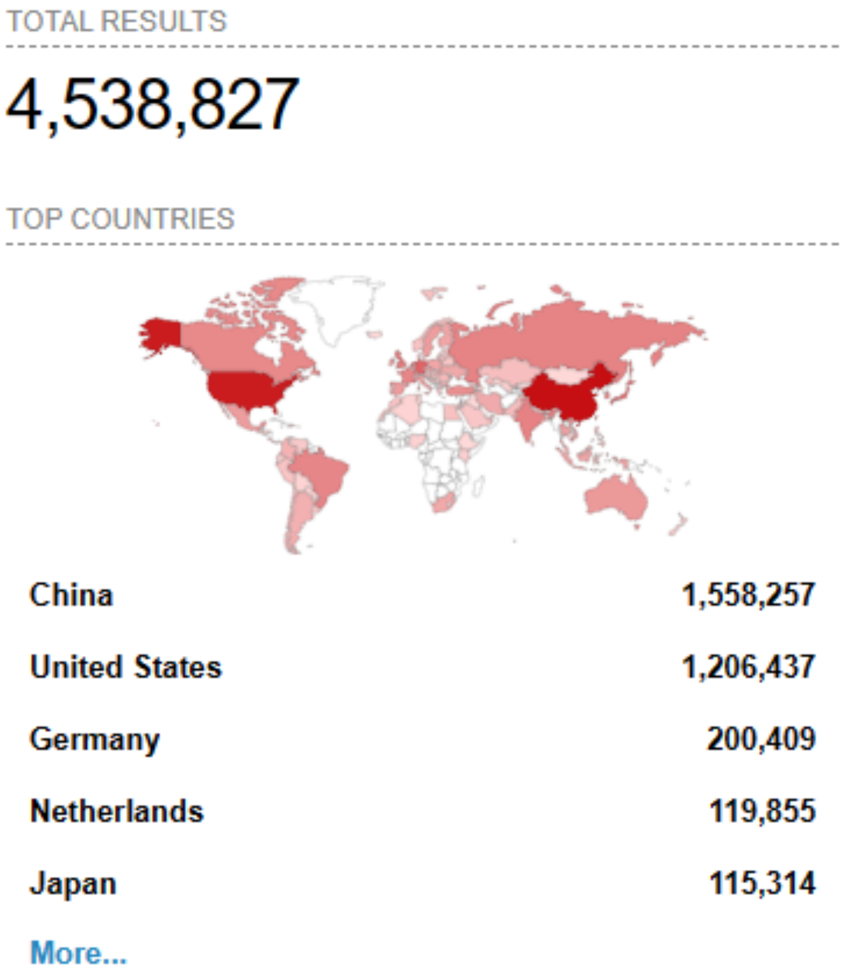
- Popular Remote Access Solution
- Legacy and Longevity
- And more?



- [MS-RDSOD]: Remote Desktop Services Protocols Overview
  - **[MS-RDSOD]: Remote Desktop Services Protocols Overview**
  - 1 Introduction
  - 2 Functional Architecture



1/31/2013	2.0	None
10/25/2012	2.0	Major
7/12/2012	1.0	None
3/30/2012	1.0	New



TOTAL RESULTS

4,538,827

TOP COUNTRIES

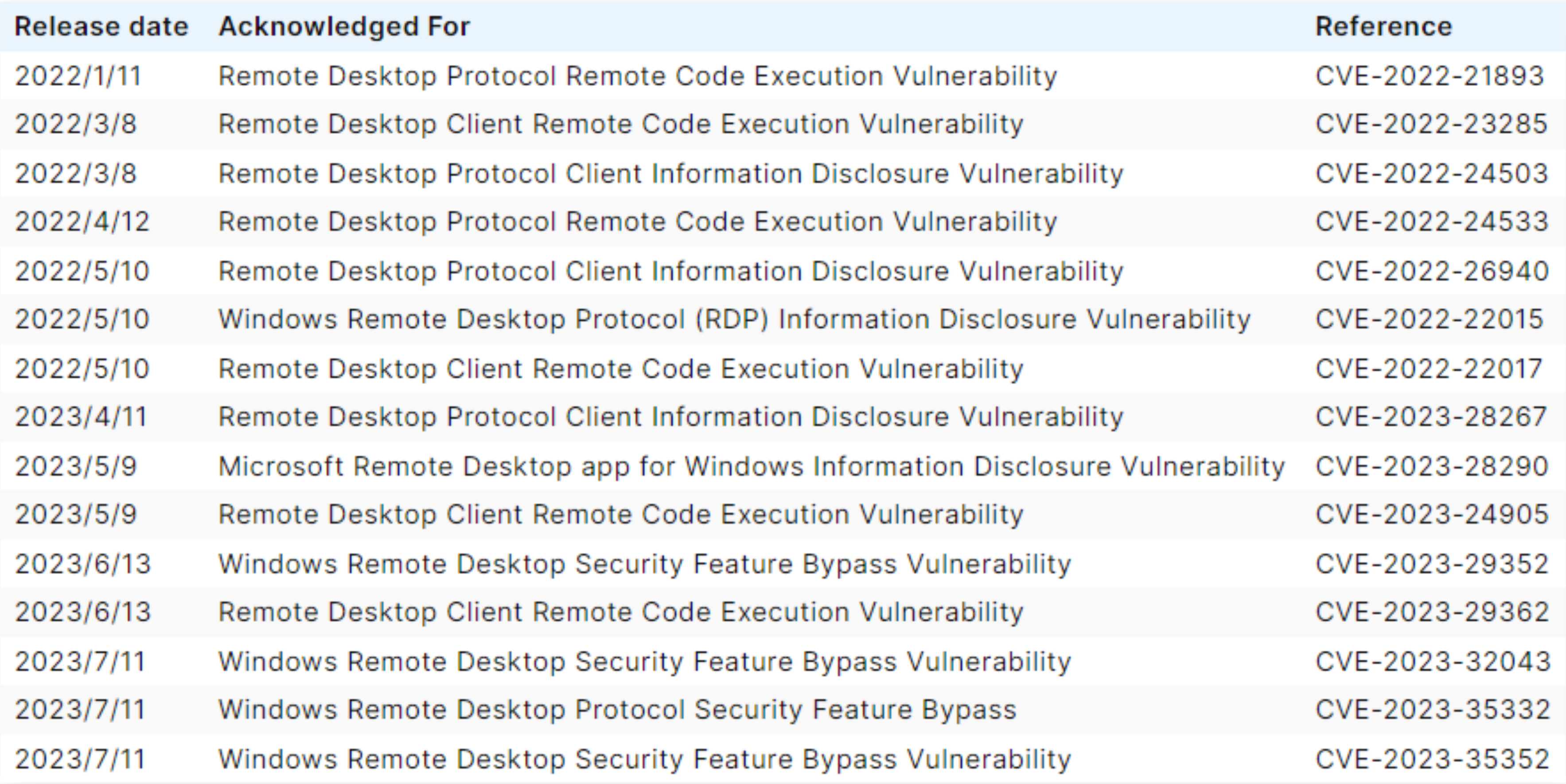
China	1,558,257
United States	1,206,437
Germany	200,409
Netherlands	119,855
Japan	115,314

More...

https://www.shodan.io/search?query=port%3A%223389%22

Motivation

- Few vulnerabilities in RDP in the past year (01/2022-09/2023)



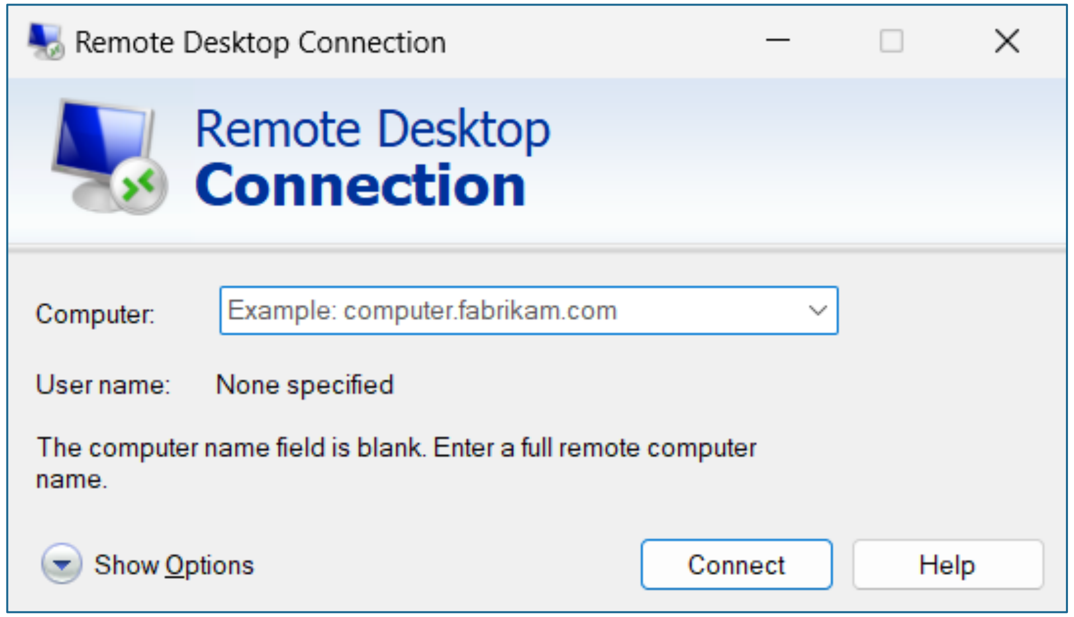
Release date	Acknowledged For	Reference
2022/1/11	Remote Desktop Protocol Remote Code Execution Vulnerability	CVE-2022-21893
2022/3/8	Remote Desktop Client Remote Code Execution Vulnerability	CVE-2022-23285
2022/3/8	Remote Desktop Protocol Client Information Disclosure Vulnerability	CVE-2022-24503
2022/4/12	Remote Desktop Protocol Remote Code Execution Vulnerability	CVE-2022-24533
2022/5/10	Remote Desktop Protocol Client Information Disclosure Vulnerability	CVE-2022-26940
2022/5/10	Windows Remote Desktop Protocol (RDP) Information Disclosure Vulnerability	CVE-2022-22015
2022/5/10	Remote Desktop Client Remote Code Execution Vulnerability	CVE-2022-22017
2023/4/11	Remote Desktop Protocol Client Information Disclosure Vulnerability	CVE-2023-28267
2023/5/9	Microsoft Remote Desktop app for Windows Information Disclosure Vulnerability	CVE-2023-28290
2023/5/9	Remote Desktop Client Remote Code Execution Vulnerability	CVE-2023-24905
2023/6/13	Windows Remote Desktop Security Feature Bypass Vulnerability	CVE-2023-29352
2023/6/13	Remote Desktop Client Remote Code Execution Vulnerability	CVE-2023-29362
2023/7/11	Windows Remote Desktop Security Feature Bypass Vulnerability	CVE-2023-32043
2023/7/11	Windows Remote Desktop Protocol Security Feature Bypass	CVE-2023-35332
2023/7/11	Windows Remote Desktop Security Feature Bypass Vulnerability	CVE-2023-35352

https://msrc.microsoft.com/report/vulnerability

Introduction

RDP Overview

- RDP contains the following features
  - **Clipboard**
  - **Printer**
  - **Storage Device**
  - **Smart Card**
  - **Audio IN/OUT**
  - …



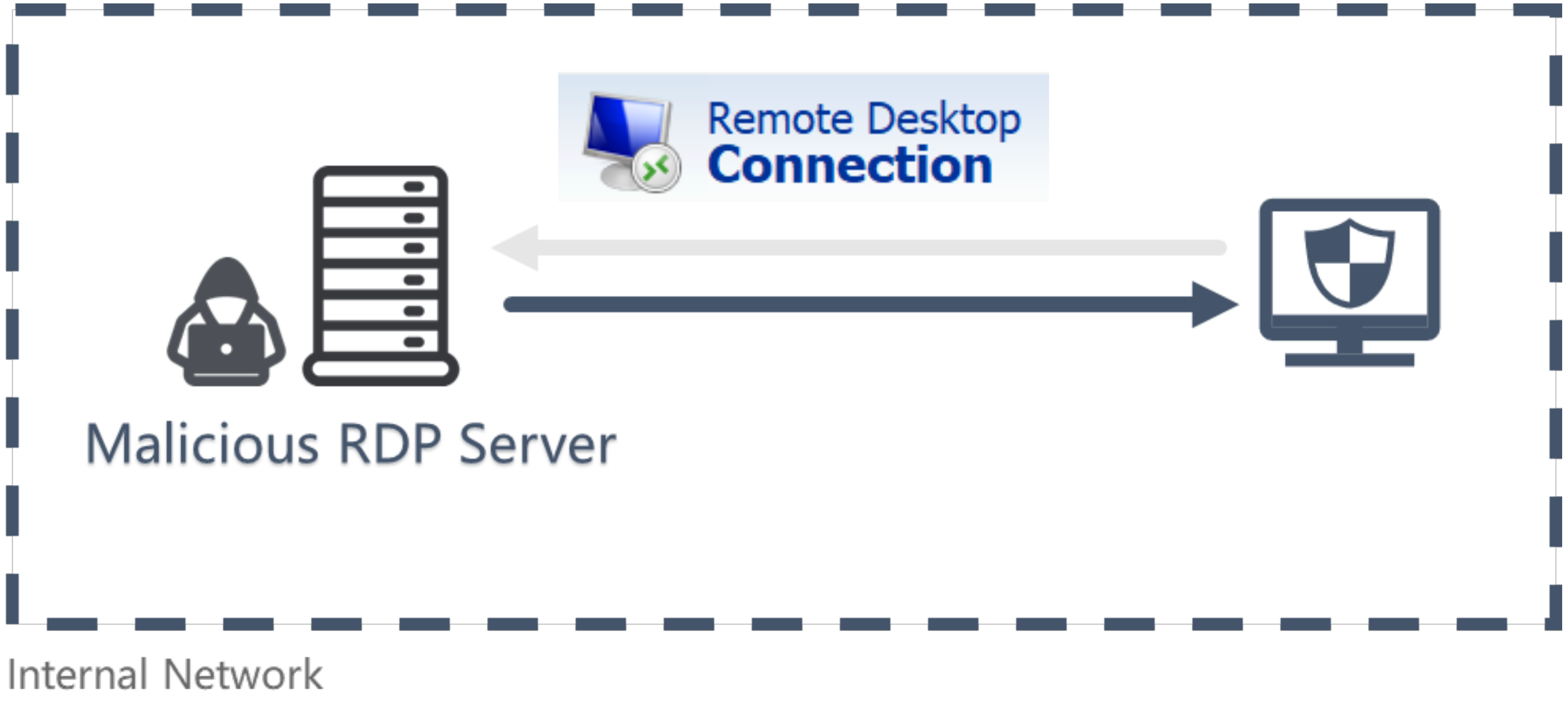
Computer: Example: computer.fabrikam.com

User name: None specified

The computer name field is blank. Enter a full remote computer name.

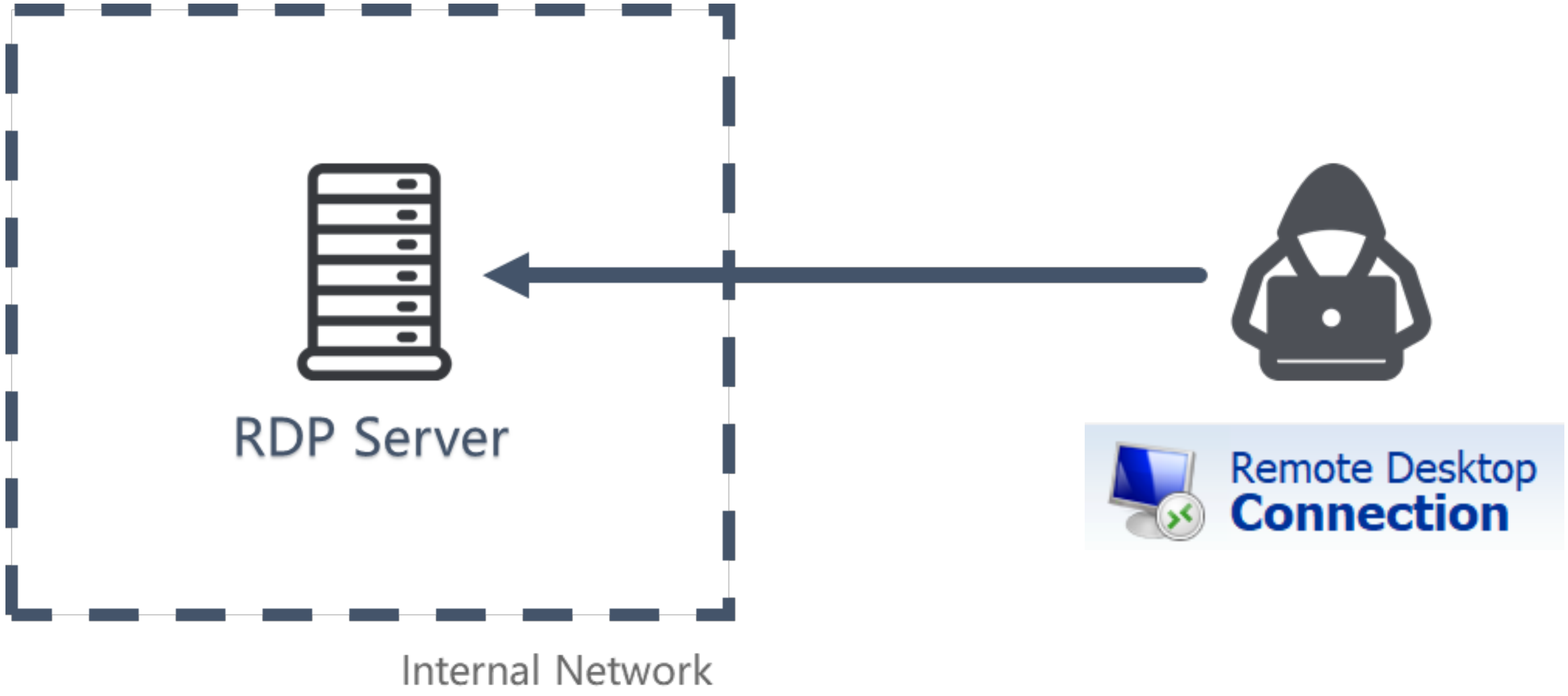
RDP Client Attack

- Victims connect malicious server using mstsc.exe



RDP Server Attack

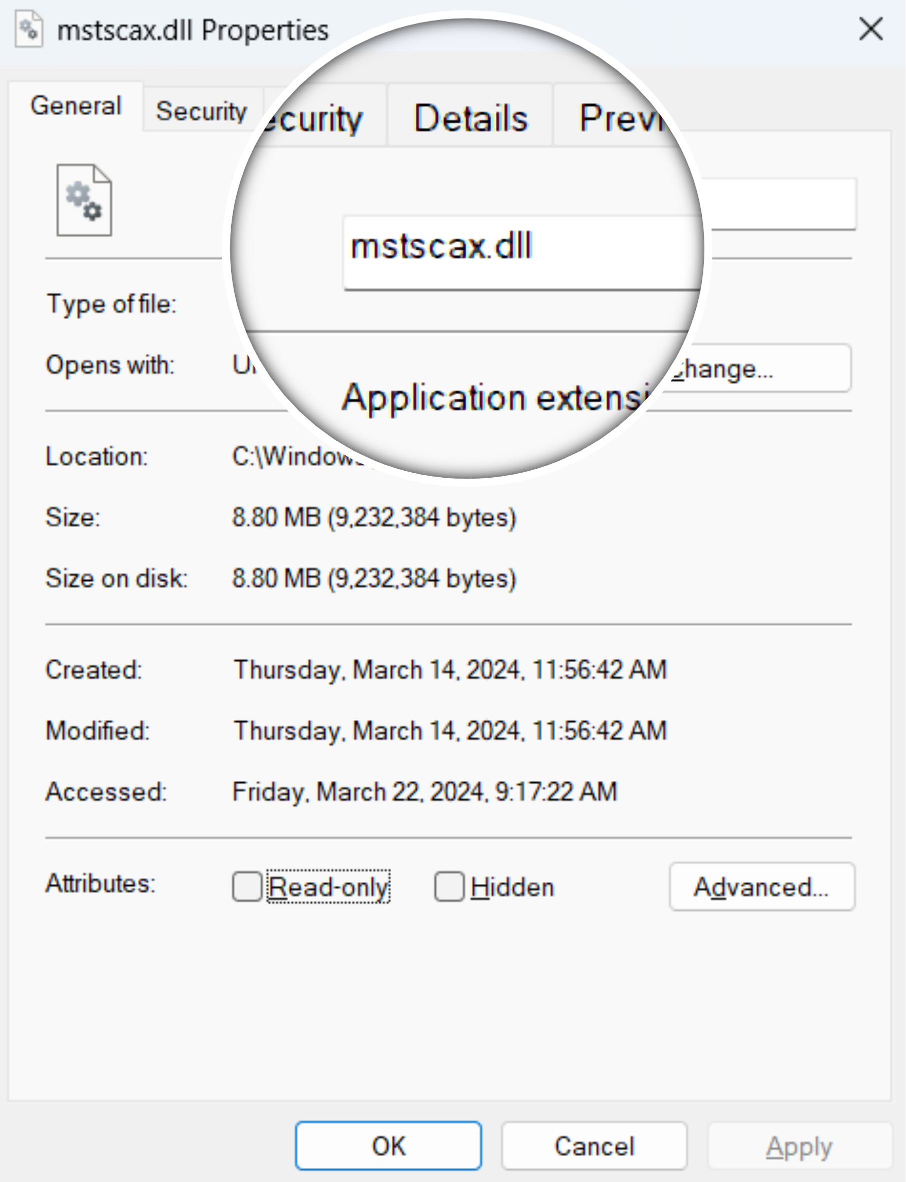
- Attackers take control of the RDP Server using mstsc.exe



Client or Server ?

Focus on Microsoft RDP Client

- Why MS RDP Client ?
  - **Clarity** (mstscax.dll, etc.)
  - **Operability** (Public APIs)
  - **Simplicity** (Compared to RDP Server)
  - **Quickly** (Learn from previous works)



mstscax.dll Properties

mstscax.dll

Type of file: Application extensi

Opens with: U

Location: C:\Window

Size: 8.80 MB (9,232,384 bytes)

Size on disk: 8.80 MB (9,232,384 bytes)

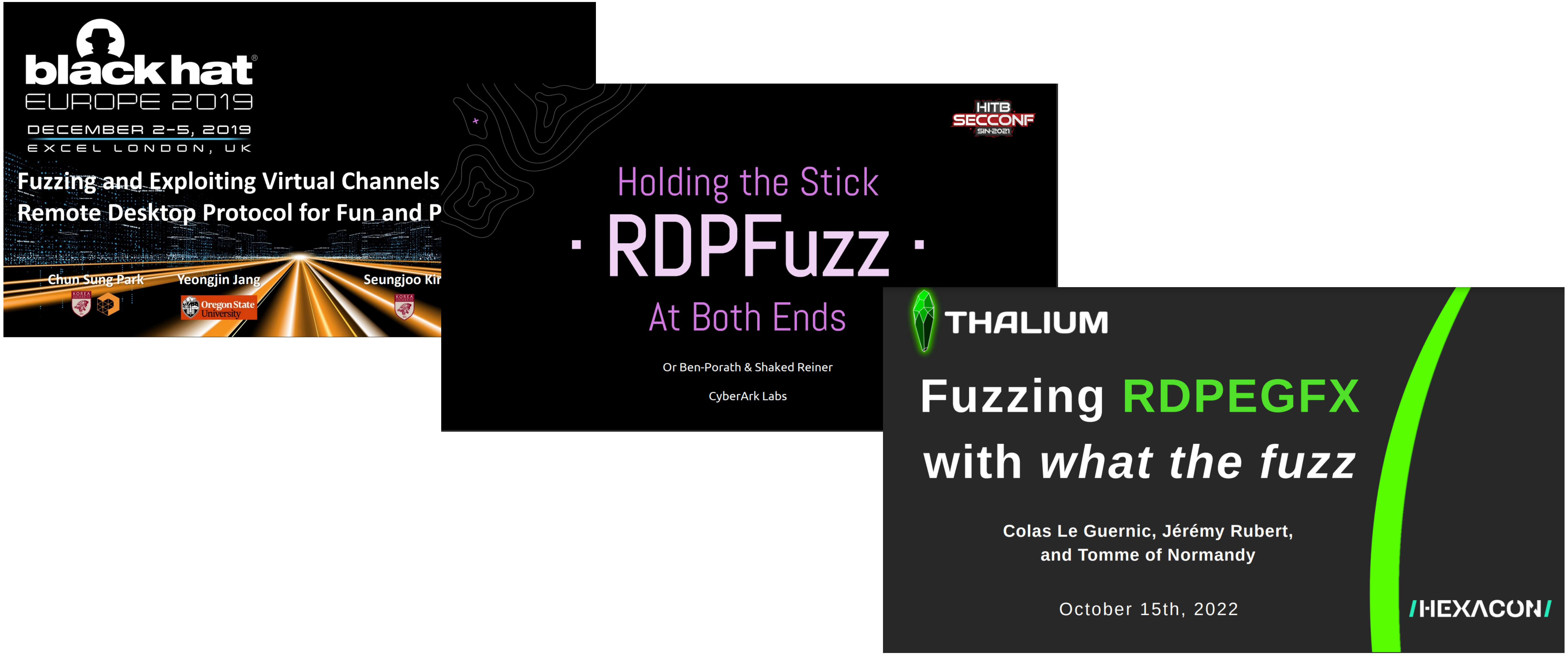
Created: Thursday, March 14, 2024, 11:56:42 AM

Modified: Thursday, March 14, 2024, 11:56:42 AM

Accessed: Friday, March 22, 2024, 9:17:22 AM

Attributes: □ Read-only □ Hidden

Previous Works



black hat EUROPE 2019

DECEMBER 2-5, 2019
EXCEL LONDON, UK

Fuzzing and Exploiting Virtual Channels
Remote Desktop Protocol for Fun and P

Chun Sung Park Yeongjin Jang Seungjoo Kir

Oregon State University

HITB SECCONF SIN-2021

Holding the Stick

· RDPFuzz ·

At Both Ends

Or Ben-Porath & Shaked Reiner

CyberArk Labs

THALIUM

Fuzzing RDPEGFX
with *what the fuzz*

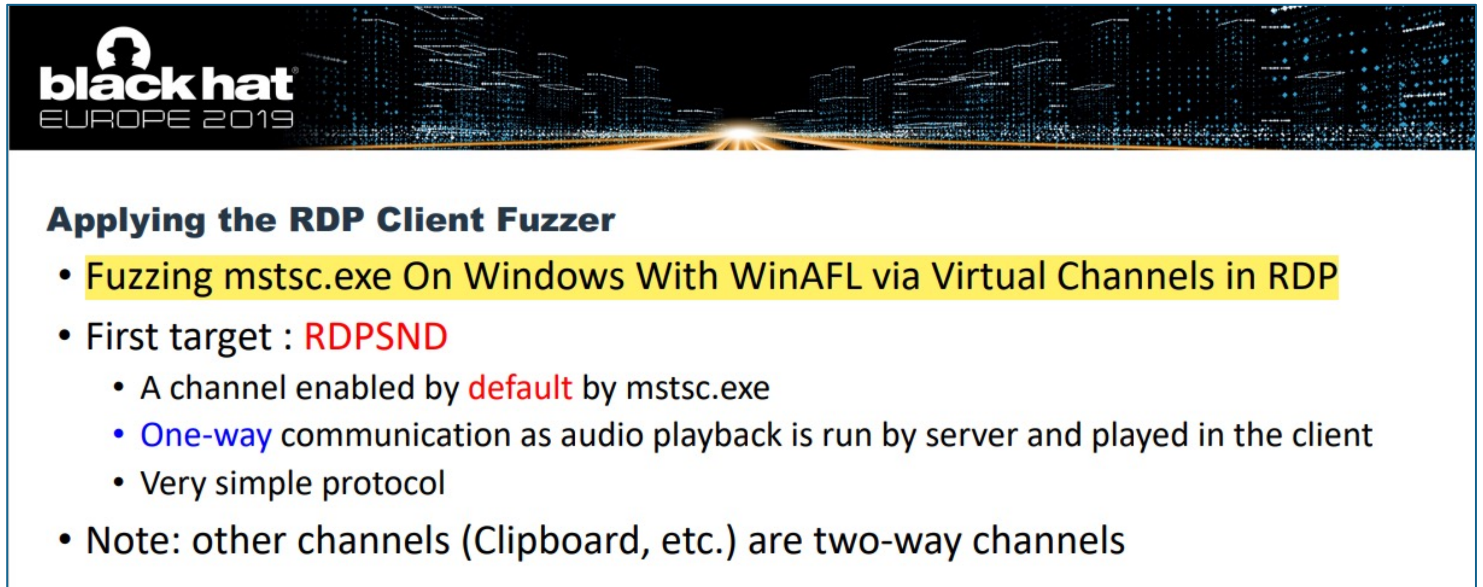
Colas Le Guernic, Jérémy Rubert,
and Tomme of Normandy

October 15th, 2022

/HEXACON/

RDP Virtual Channel

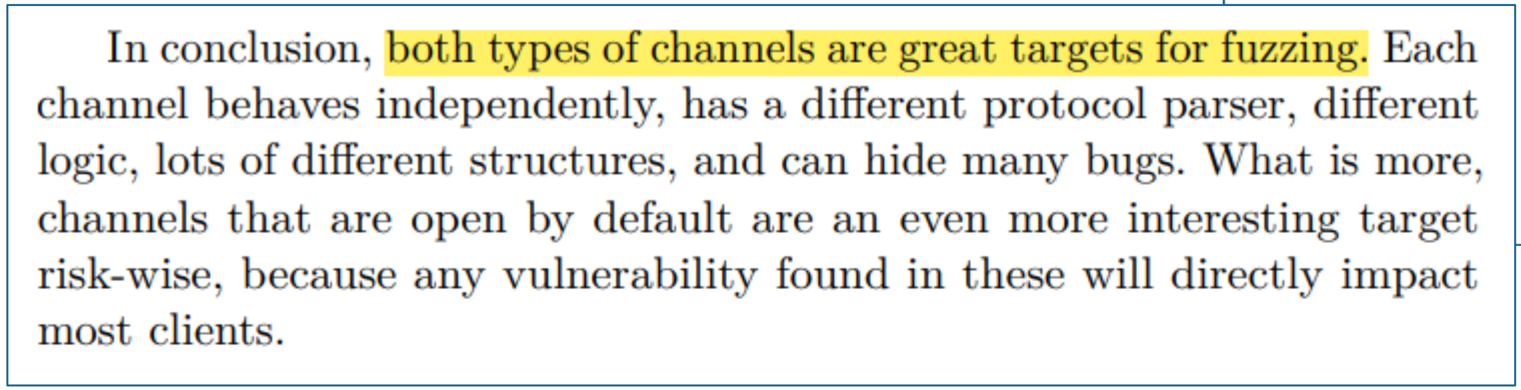
- Virtual Channel
  - **Static Virtual Channel**
  - **Dynamic Virtual Channel**



Applying the RDP Client Fuzzer

- Fuzzing mstsc.exe On Windows With WinAFL via Virtual Channels in RDP
- First target : RDPSND
  - A channel enabled by default by mstsc.exe
  - One-way communication as audio playback is run by server and played in the client
  - Very simple protocol
- Note: other channels (Clipboard, etc.) are two-way channels

#BHEU @BLACK HAT EVENTS

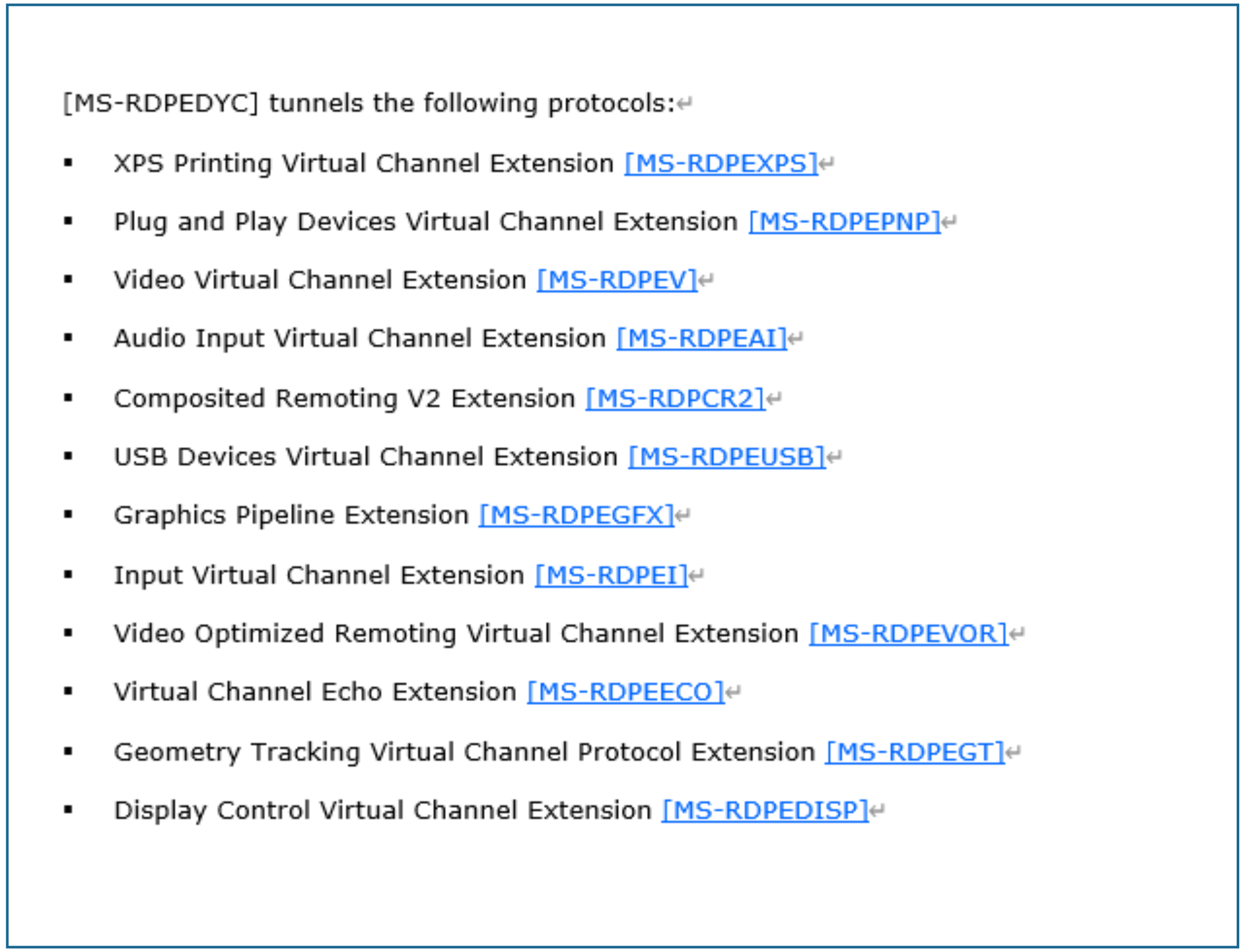


In conclusion, both types of channels are great targets for fuzzing. Each channel behaves independently, has a different protocol parser, different logic, lots of different structures, and can hide many bugs. What is more, channels that are open by default are an even more interesting target risk-wise, because any vulnerability found in these will directly impact most clients.

https://www.blackhat.com/eu-19/briefings/schedule/#fuzzing-and-exploiting-virtual-channels-in-microsoft-remote-desktop-protocol-for-fun-and-profit-17789

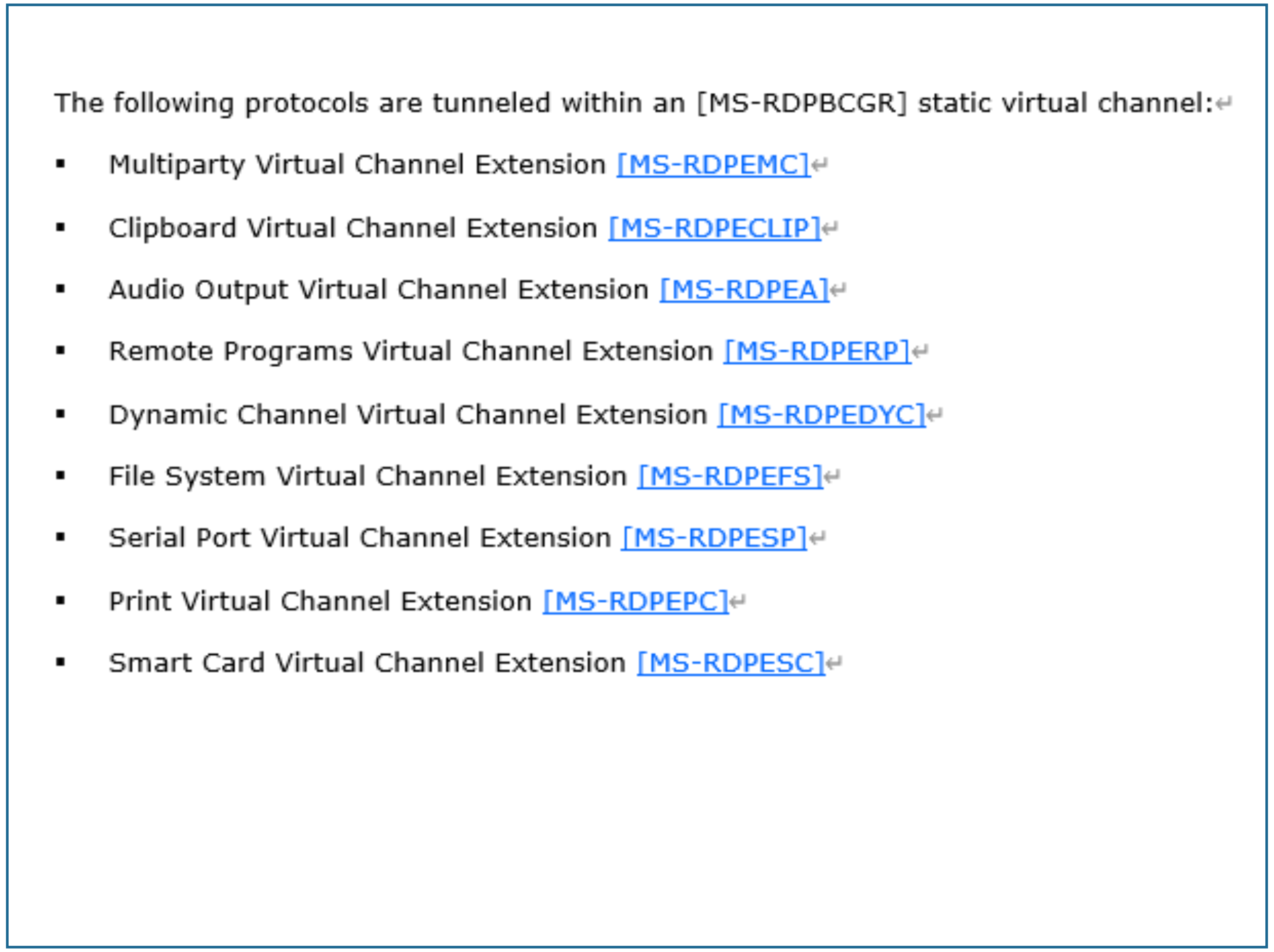
https://www.sstic.org/media/SSTIC2022/SSTIC-actes/fuzzing_microsofts_rdp_client_using_virtual_channe/SSTIC2022-Article-fuzzing_microsofts_rdp_client_using_virtual_channels-ricotta.pdf

RDP Virtual Channel



[MS-RDPEDYC] tunnels the following protocols:

- XPS Printing Virtual Channel Extension [MS-RDPEXPS]
- Plug and Play Devices Virtual Channel Extension [MS-RDPEPNP]
- Video Virtual Channel Extension [MS-RDPEV]
- Audio Input Virtual Channel Extension [MS-RDPEAI]
- Composited Remoting V2 Extension [MS-RDPCR2]
- USB Devices Virtual Channel Extension [MS-RDPEUSB]
- Graphics Pipeline Extension [MS-RDPEGFX]
- Input Virtual Channel Extension [MS-RDPEI]
- Video Optimized Remoting Virtual Channel Extension [MS-RDPEVOR]
- Virtual Channel Echo Extension [MS-RDPEECO]
- Geometry Tracking Virtual Channel Protocol Extension [MS-RDPEGT]
- Display Control Virtual Channel Extension [MS-RDPEDISP]



The following protocols are tunneled within an [MS-RDPBCGR] static virtual channel:

- Multiparty Virtual Channel Extension [MS-RDPEMC]
- Clipboard Virtual Channel Extension [MS-RDPECLIP]
- Audio Output Virtual Channel Extension [MS-RDPEA]
- Remote Programs Virtual Channel Extension [MS-RDPERP]
- Dynamic Channel Virtual Channel Extension [MS-RDPEDYC]
- File System Virtual Channel Extension [MS-RDPEFS]
- Serial Port Virtual Channel Extension [MS-RDPESP]
- Print Virtual Channel Extension [MS-RDPEPC]
- Smart Card Virtual Channel Extension [MS-RDPESC]

RDP Virtual Channel

RDPSND

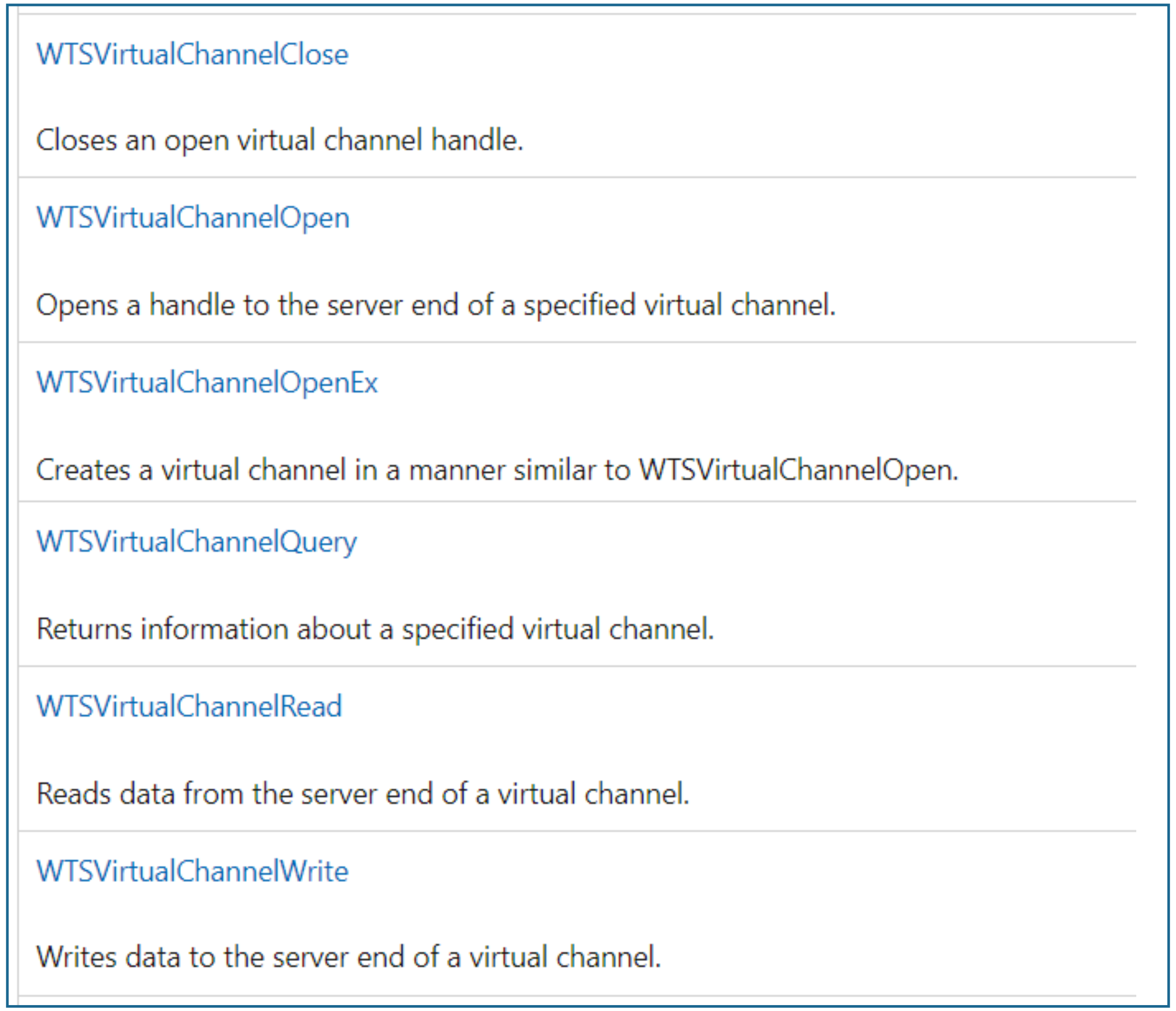
RDPDR

TSMF

...

Virtual Channel API

- WTS API
  - Open Server
  - Open Virtual Channel
  - **Write / Read Virtual Channel**
  - Close Virtual Channel
  - Close Server
  - …



WTSVirtualChannelClose

Closes an open virtual channel handle.

WTSVirtualChannelOpen

Opens a handle to the server end of a specified virtual channel.

WTSVirtualChannelOpenEx

Creates a virtual channel in a manner similar to WTSVirtualChannelOpen.

WTSVirtualChannelQuery

Returns information about a specified virtual channel.

WTSVirtualChannelRead

Reads data from the server end of a virtual channel.

WTSVirtualChannelWrite

Writes data to the server end of a virtual channel.

https://learn.microsoft.com/en-us/windows/win32/api/wtsapi32/

Fuzzing

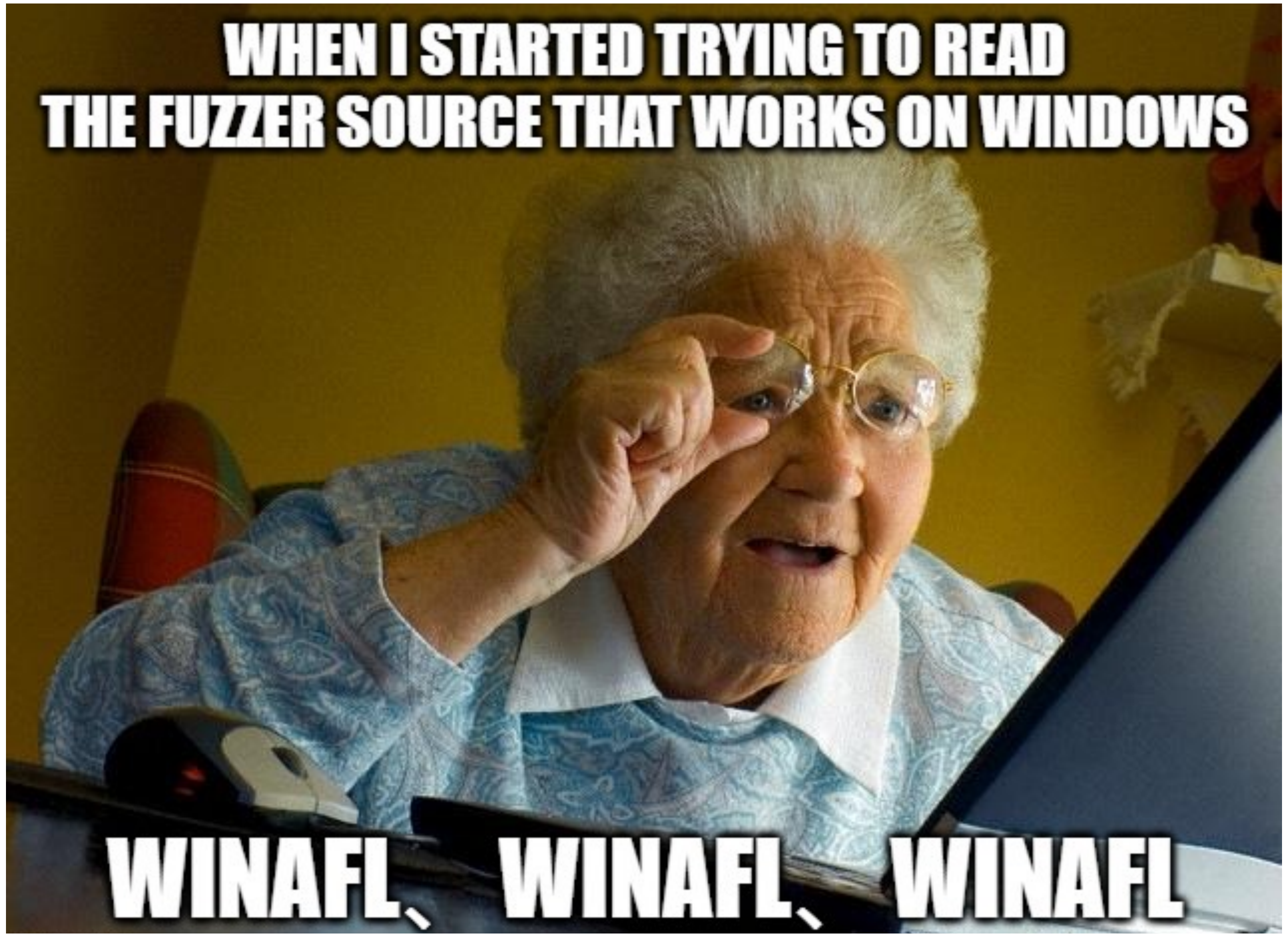
Open Source RDP Fuzzer

rdpfuzz

- https://github.com/cyberark/rdpfuzz

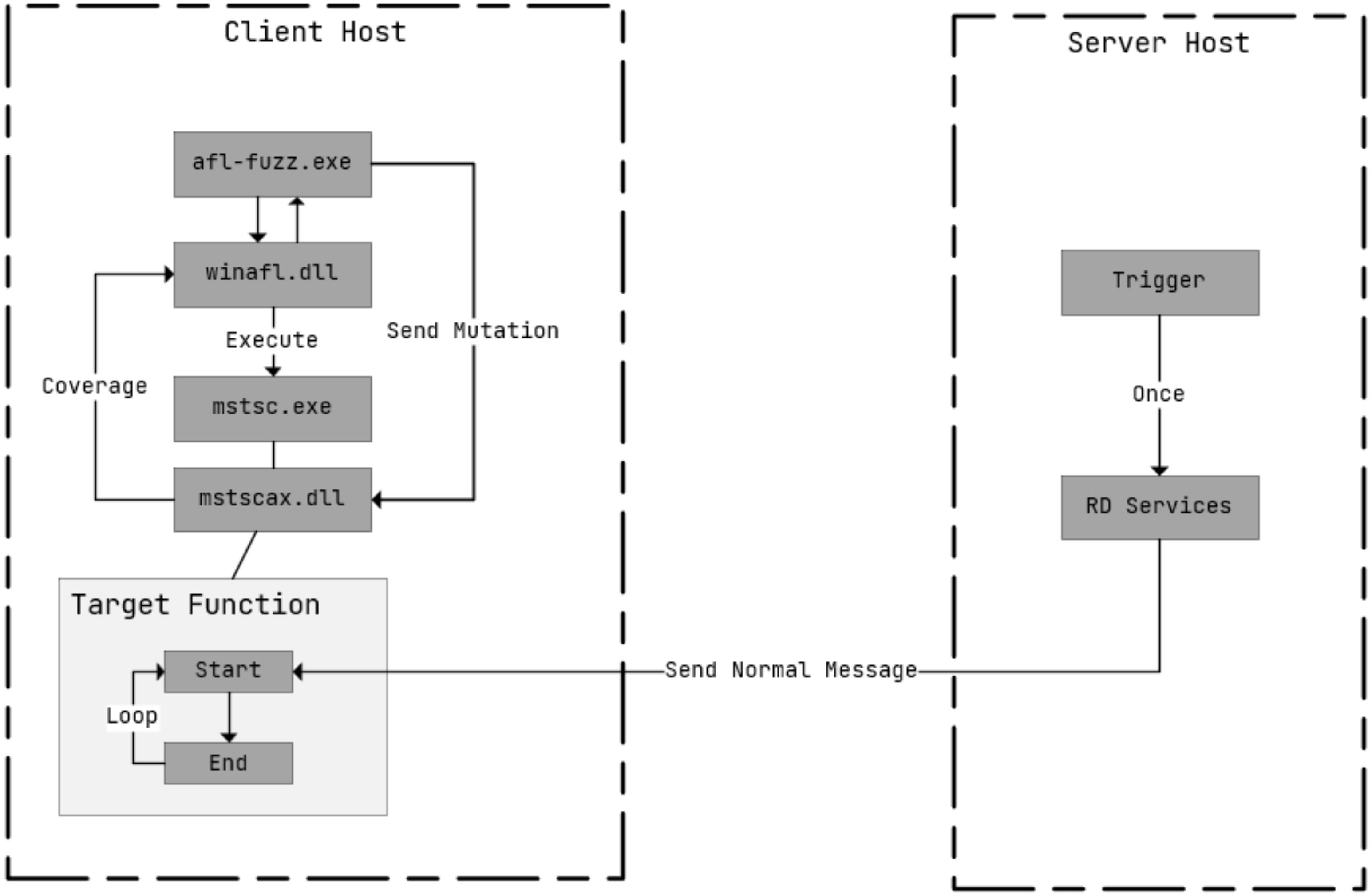
WinAFL-RDP

- https://github.com/Team-BT5/WinAFL-RDP


WHEN I STARTED TRYING TO READ
THE FUZZER SOURCE THAT WORKS ON WINDOWS
WINAFL、 WINAFL、 WINAFL

Fuzzing Architecture #1

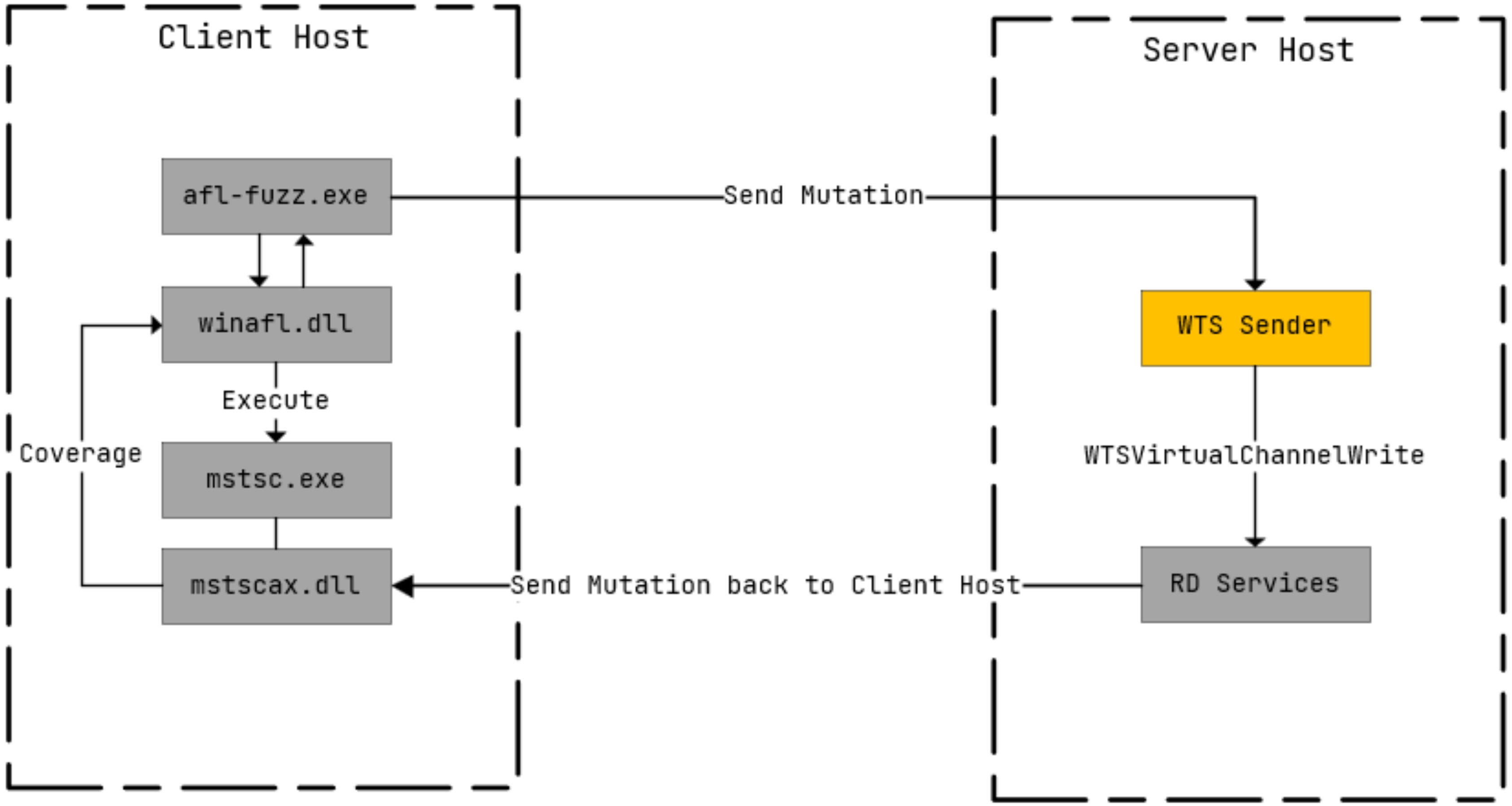
- Loop



https://github.com/Team-BT5/WinAFL-RDP

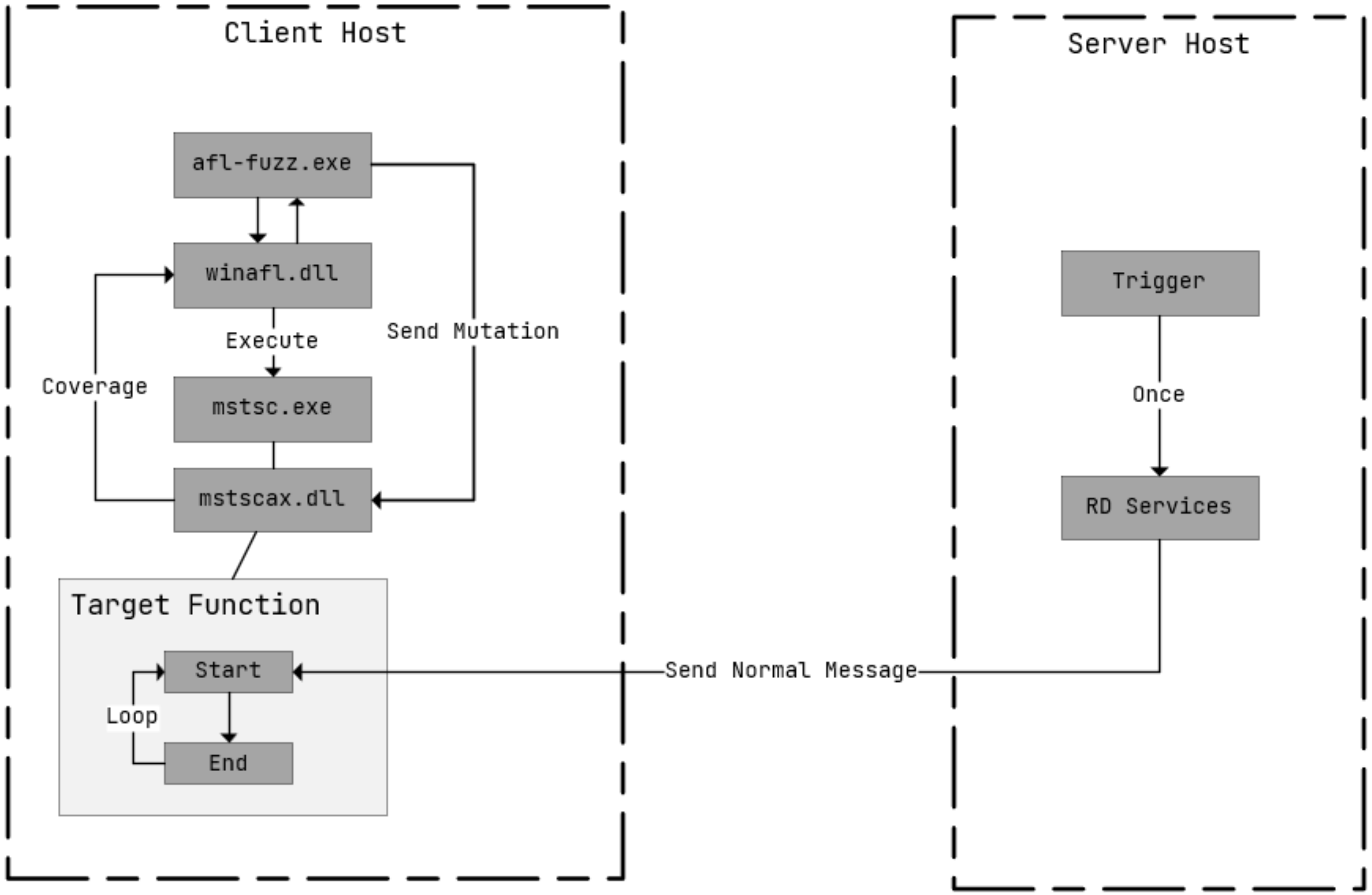
Fuzzing Architecture #2

- Proxy



https://github.com/cyberark/rdpfuzz

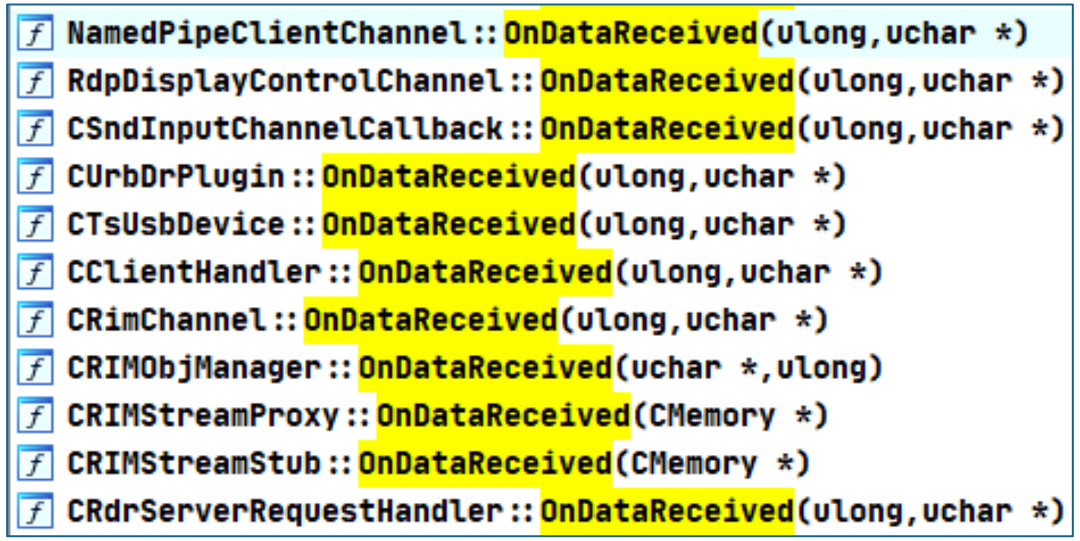
Choose Fuzzer



https://github.com/Team-BT5/WinAFL-RDP

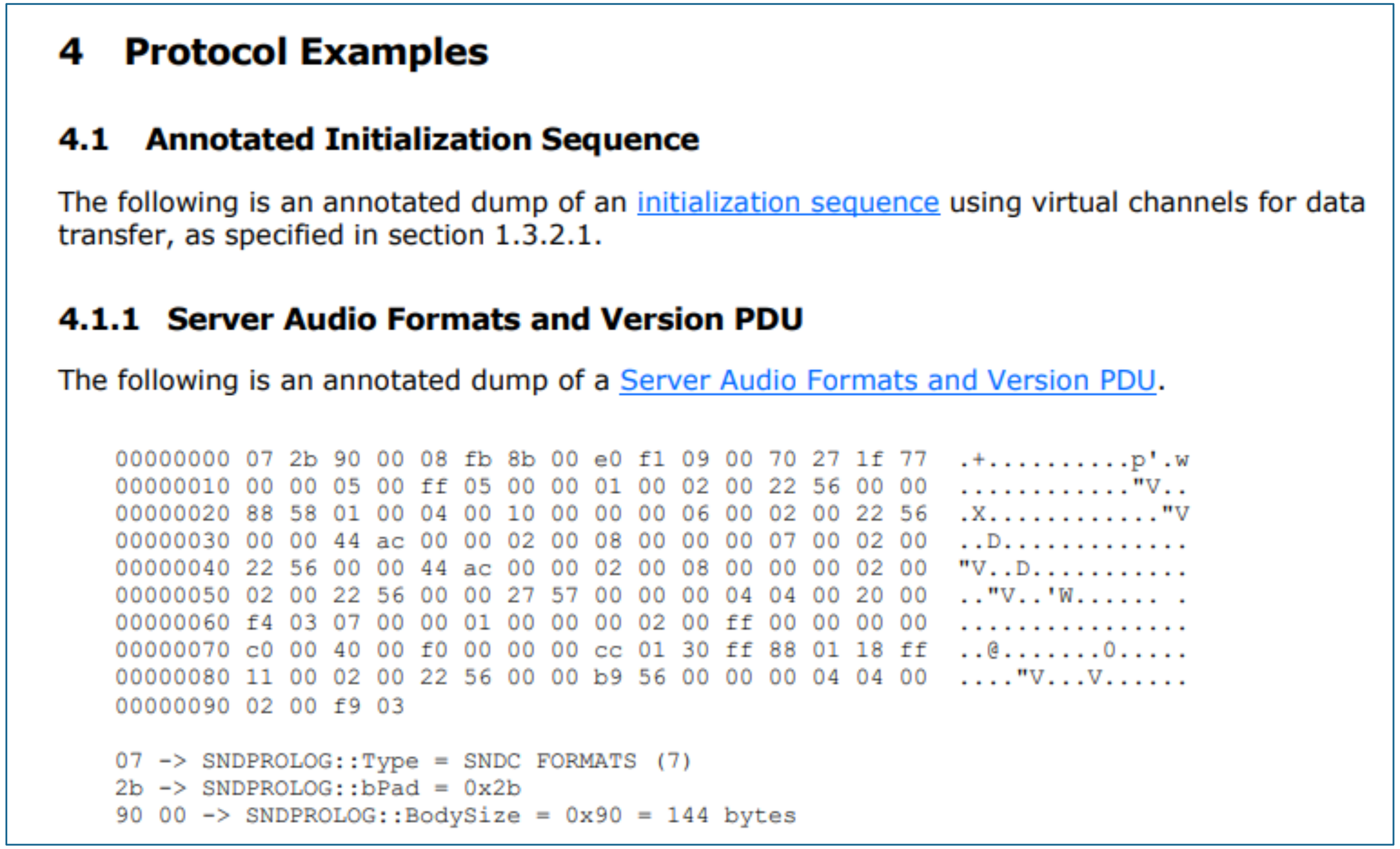
Before Fuzzing

- **Target**
- **Seeds**



```
NamedPipeClientChannel::OnDataReceived(ulong,uchar *)
RdpDisplayControlChannel::OnDataReceived(ulong,uchar *)
CSndInputChannelCallback::OnDataReceived(ulong,uchar *)
CUrbDrPlugin::OnDataReceived(ulong,uchar *)
CTsUsbDevice::OnDataReceived(ulong,uchar *)
CClientHandler::OnDataReceived(ulong,uchar *)
CRimChannel::OnDataReceived(ulong,uchar *)
CRIMObjManager::OnDataReceived(uchar *,ulong)
CRIMStreamProxy::OnDataReceived(CMemory *)
CRIMStreamStub::OnDataReceived(CMemory *)
CRdrServerRequestHandler::OnDataReceived(ulong,uchar *)
```

Regular Expr: **.*::OnDataReceived**



4 Protocol Examples

4.1 Annotated Initialization Sequence

The following is an annotated dump of an initialization sequence using virtual channels for data transfer, as specified in section 1.3.2.1.

4.1.1 Server Audio Formats and Version PDU

The following is an annotated dump of a Server Audio Formats and Version PDU.

```
00000000 07 2b 90 00 08 fb 8b 00 e0 f1 09 00 70 27 1f 77  .+..........p'.w
00000010 00 00 05 00 ff 05 00 00 01 00 02 00 22 56 00 00  ............"V..
00000020 88 58 01 00 04 00 10 00 00 00 06 00 02 00 22 56  .X............"V
00000030 00 00 44 ac 00 00 02 00 08 00 00 00 07 00 02 00  ..D.............
00000040 22 56 00 00 44 ac 00 00 02 00 08 00 00 00 02 00  "V..D...........
00000050 02 00 22 56 00 00 27 57 00 00 00 04 04 00 20 00  .."V..'W...... .
00000060 f4 03 07 00 00 01 00 00 00 02 00 ff 00 00 00 00  ................
00000070 c0 00 40 00 f0 00 00 00 cc 01 30 ff 88 01 18 ff  ..@.......0.....
00000080 11 00 02 00 22 56 00 00 b9 56 00 00 00 04 04 00  ...."V...V......
00000090 02 00 f9 03

07 -> SNDPROLOG::Type = SNDC FORMATS (7)
2b -> SNDPROLOG::bPad = 0x2b
90 00 -> SNDPROLOG::BodySize = 0x90 = 144 bytes
```

Environment Preparation

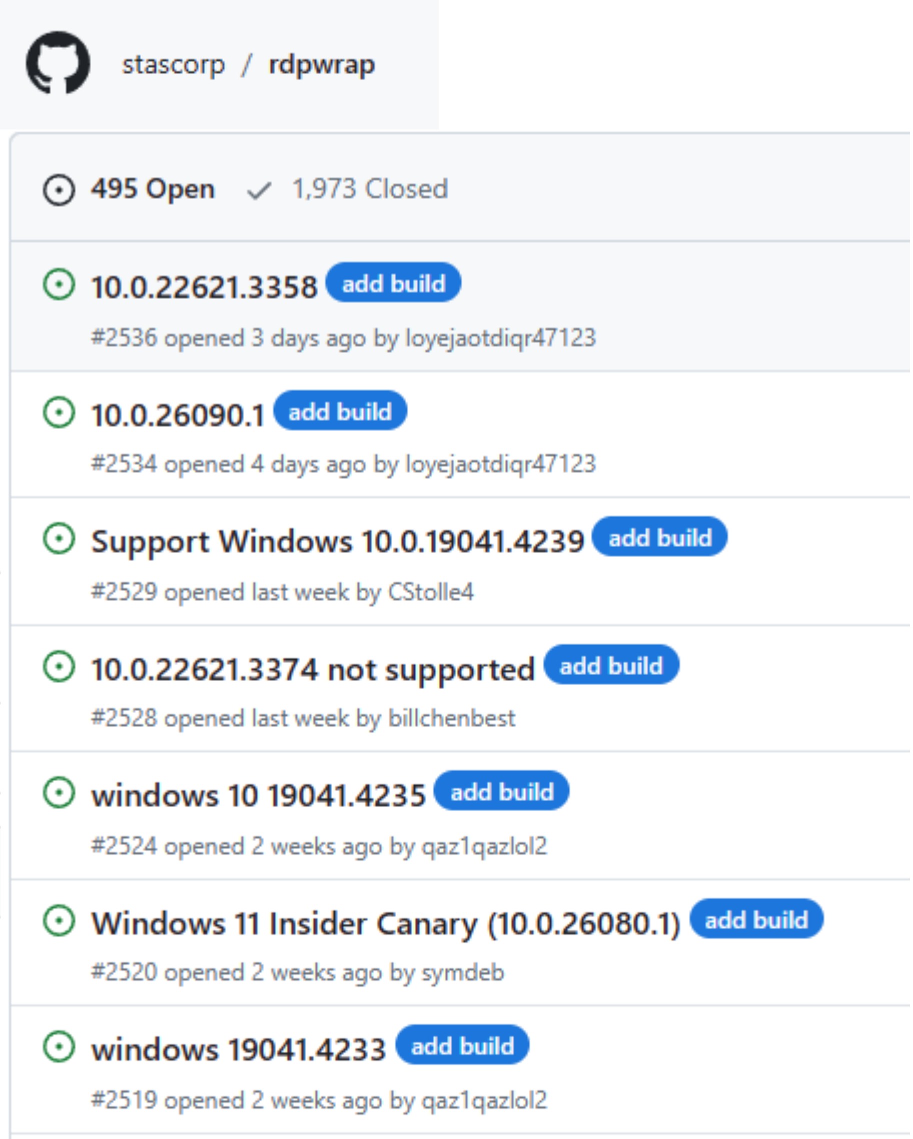
- 2 Virtual Machines
- 1 Virtual Machines + RDPWrap

Environment Preparation #1

- 2 Virtual Machines
- 1 Virtual Machines + RDPWrap

Environment Preparation #1

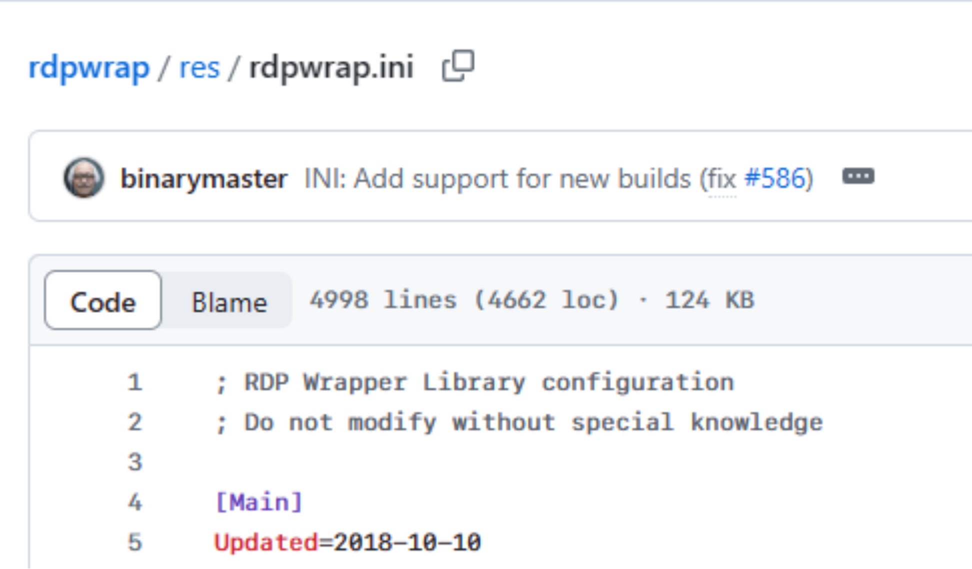
- 2 Virtual Machines
- 1 Virtual Machines + RDPWrap



stascorp / rdpwrap

495 Open ✓ 1,973 Closed

- 10.0.22621.3358 add build
  #2536 opened 3 days ago by loyejaotdiqr47123
- 10.0.26090.1 add build
  #2534 opened 4 days ago by loyejaotdiqr47123
- Support Windows 10.0.19041.4239 add build
  #2529 opened last week by CStolle4
- 10.0.22621.3374 not supported add build
  #2528 opened last week by billchenbest
- windows 10 19041.4235 add build
  #2524 opened 2 weeks ago by qaz1qazlol2
- Windows 11 Insider Canary (10.0.26080.1) add build
  #2520 opened 2 weeks ago by symdeb
- windows 19041.4233 add build
  #2519 opened 2 weeks ago by qaz1qazlol2



rdpwrap / res / rdpwrap.ini

binarymaster INI: Add support for new builds (fix #586)

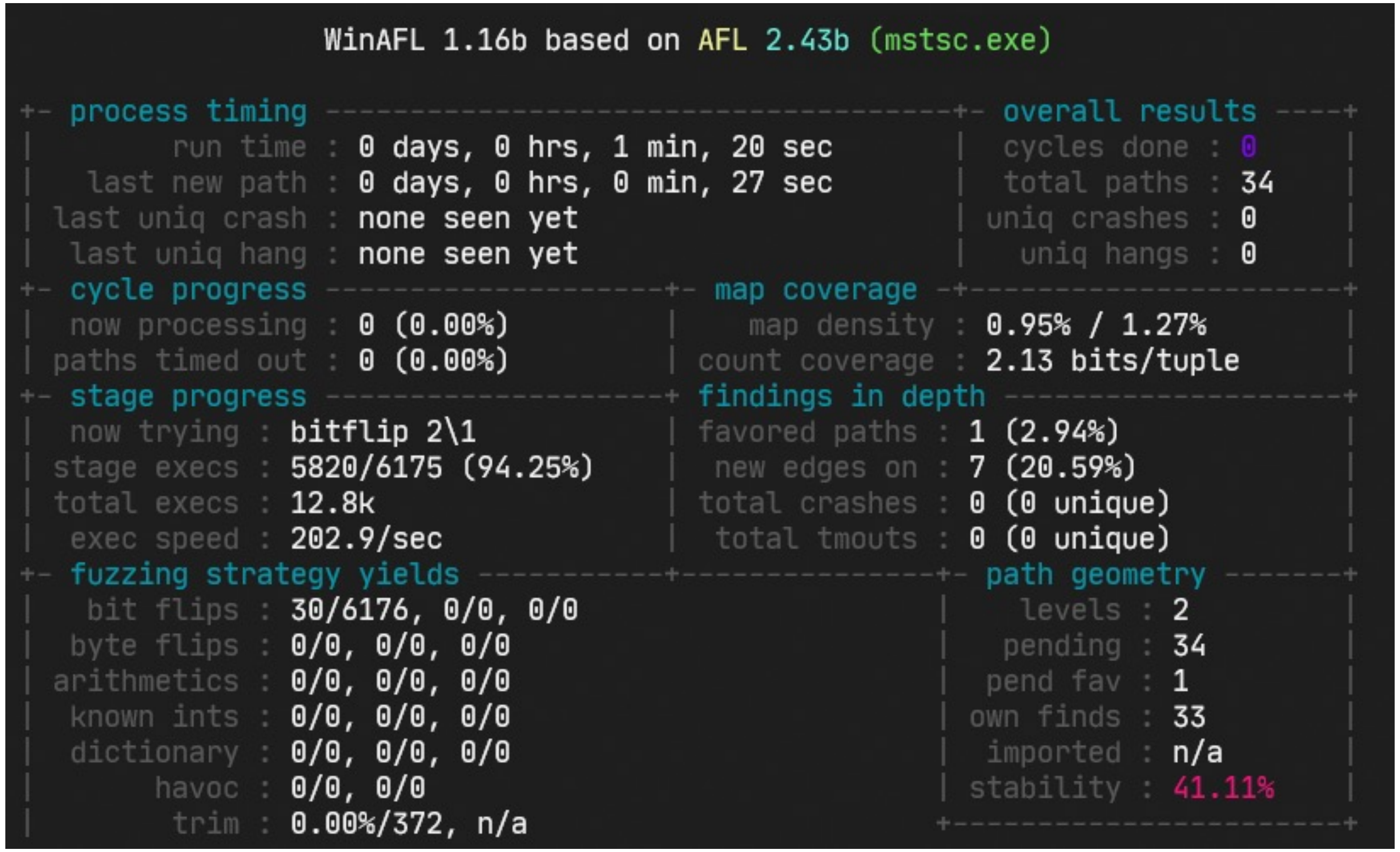
4998 lines (4662 loc) · 124 KB

```
; RDP Wrapper Library configuration
; Do not modify without special knowledge

[Main]
Updated=2018-10-10
```

Environment Preparation #1

- 2 Virtual Machines
- 1 Virtual Machines + RDPWrap

Start Fuzzing

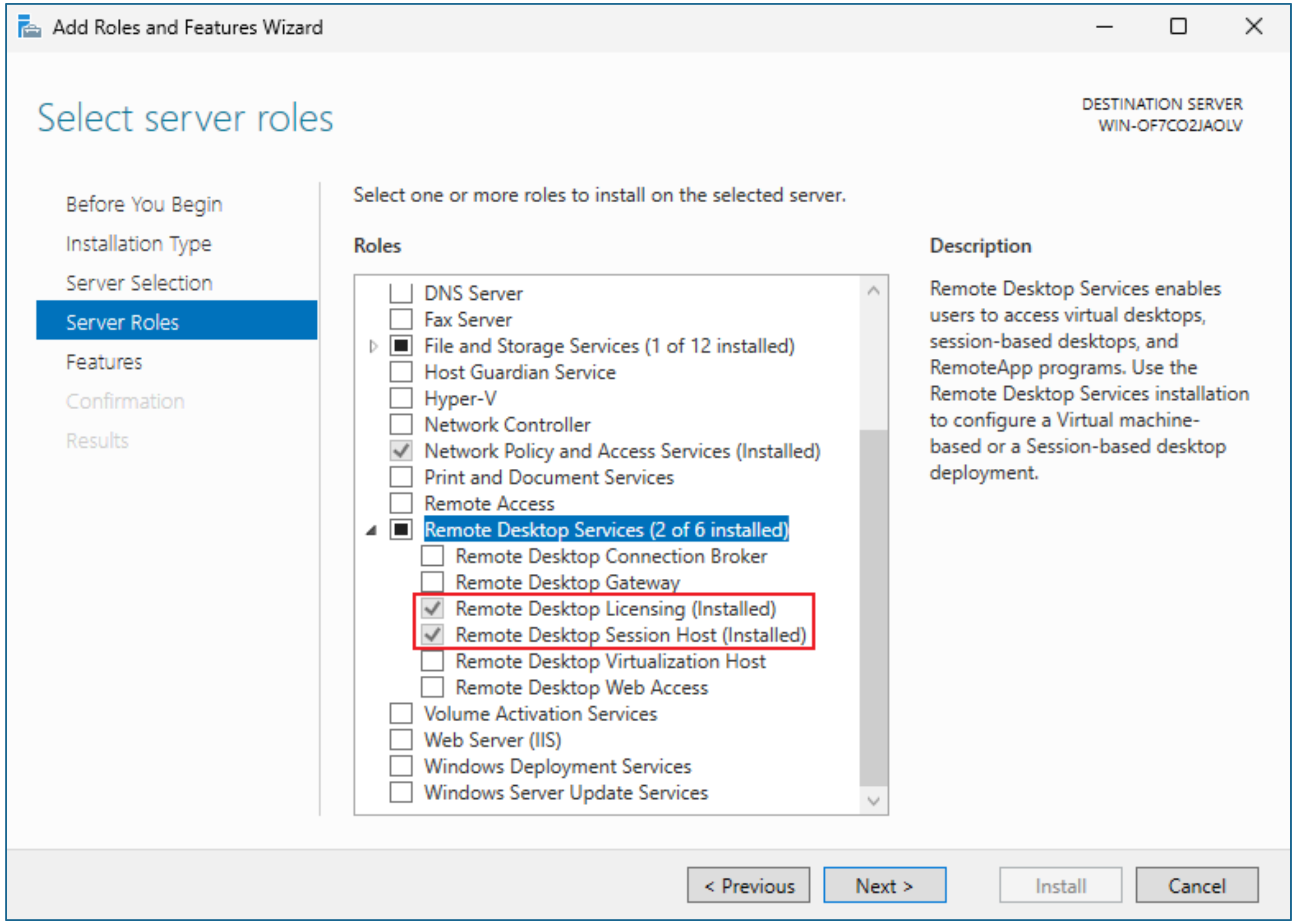


```
                   WinAFL 1.16b based on AFL 2.43b (mstsc.exe)

+- process timing -------------------------------------+- overall results ----+
|        run time : 0 days, 0 hrs, 1 min, 20 sec       |  cycles done : 0     |
|   last new path : 0 days, 0 hrs, 0 min, 27 sec       |  total paths : 34    |
| last uniq crash : none seen yet                      | uniq crashes : 0     |
|  last uniq hang : none seen yet                      |   uniq hangs : 0     |
+- cycle progress --------------------+- map coverage -+----------------------+
|  now processing : 0 (0.00%)         |    map density : 0.95% / 1.27%        |
| paths timed out : 0 (0.00%)         | count coverage : 2.13 bits/tuple      |
+- stage progress --------------------+ findings in depth --------------------+
|  now trying : bitflip 2\1           | favored paths : 1 (2.94%)             |
| stage execs : 5820/6175 (94.25%)    |  new edges on : 7 (20.59%)            |
| total execs : 12.8k                 | total crashes : 0 (0 unique)          |
|  exec speed : 202.9/sec             |  total tmouts : 0 (0 unique)          |
+- fuzzing strategy yields -----------+---------------+- path geometry -------+
|   bit flips : 30/6176, 0/0, 0/0                     |    levels : 2         |
|  byte flips : 0/0, 0/0, 0/0                         |   pending : 34        |
| arithmetics : 0/0, 0/0, 0/0                         |  pend fav : 1         |
|  known ints : 0/0, 0/0, 0/0                         | own finds : 33        |
|  dictionary : 0/0, 0/0, 0/0                         |  imported : n/a       |
|       havoc : 0/0, 0/0                              | stability : 41.11%    |
|        trim : 0.00%/372, n/a                        +-----------------------+
```

Batch Deploy

- ~~2 Virtual Machines~~
- 1 Virtual Machines + RDPWrap
- Others?

RDS (Remote Desktop Service)



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
WIN-OF7CO2JAOLV

Select one or more roles to install on the selected server.

Roles

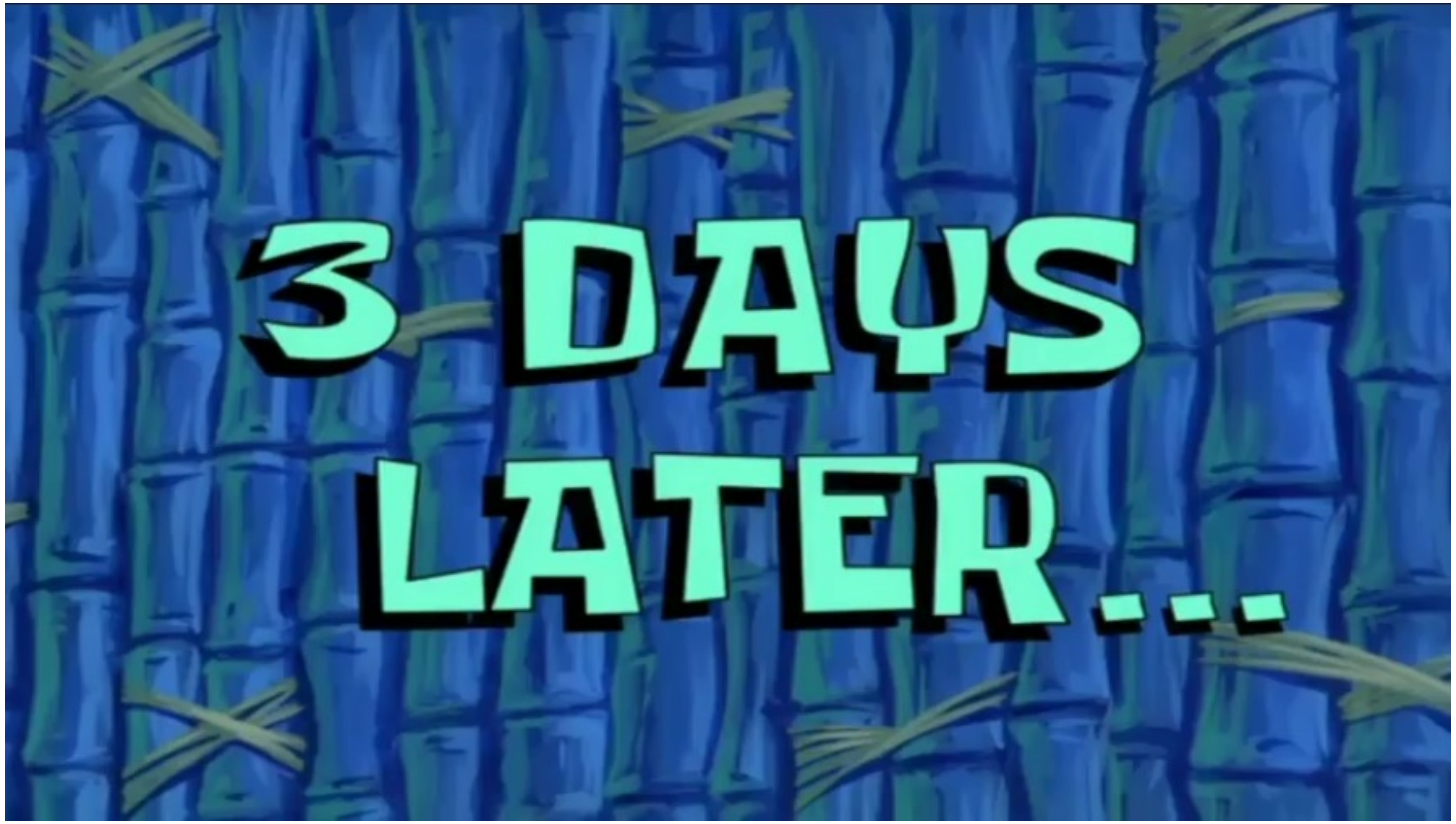
- □ DNS Server
- □ Fax Server
- ■ File and Storage Services (1 of 12 installed)
- □ Host Guardian Service
- □ Hyper-V
- □ Network Controller
- ☑ Network Policy and Access Services (Installed)
- □ Print and Document Services
- □ Remote Access
- ■ Remote Desktop Services (2 of 6 installed)
  - □ Remote Desktop Connection Broker
  - □ Remote Desktop Gateway
  - ☑ Remote Desktop Licensing (Installed)
  - ☑ Remote Desktop Session Host (Installed)
  - □ Remote Desktop Virtualization Host
  - □ Remote Desktop Web Access
- □ Volume Activation Services
- □ Web Server (IIS)
- □ Windows Deployment Services
- □ Windows Server Update Services

Description

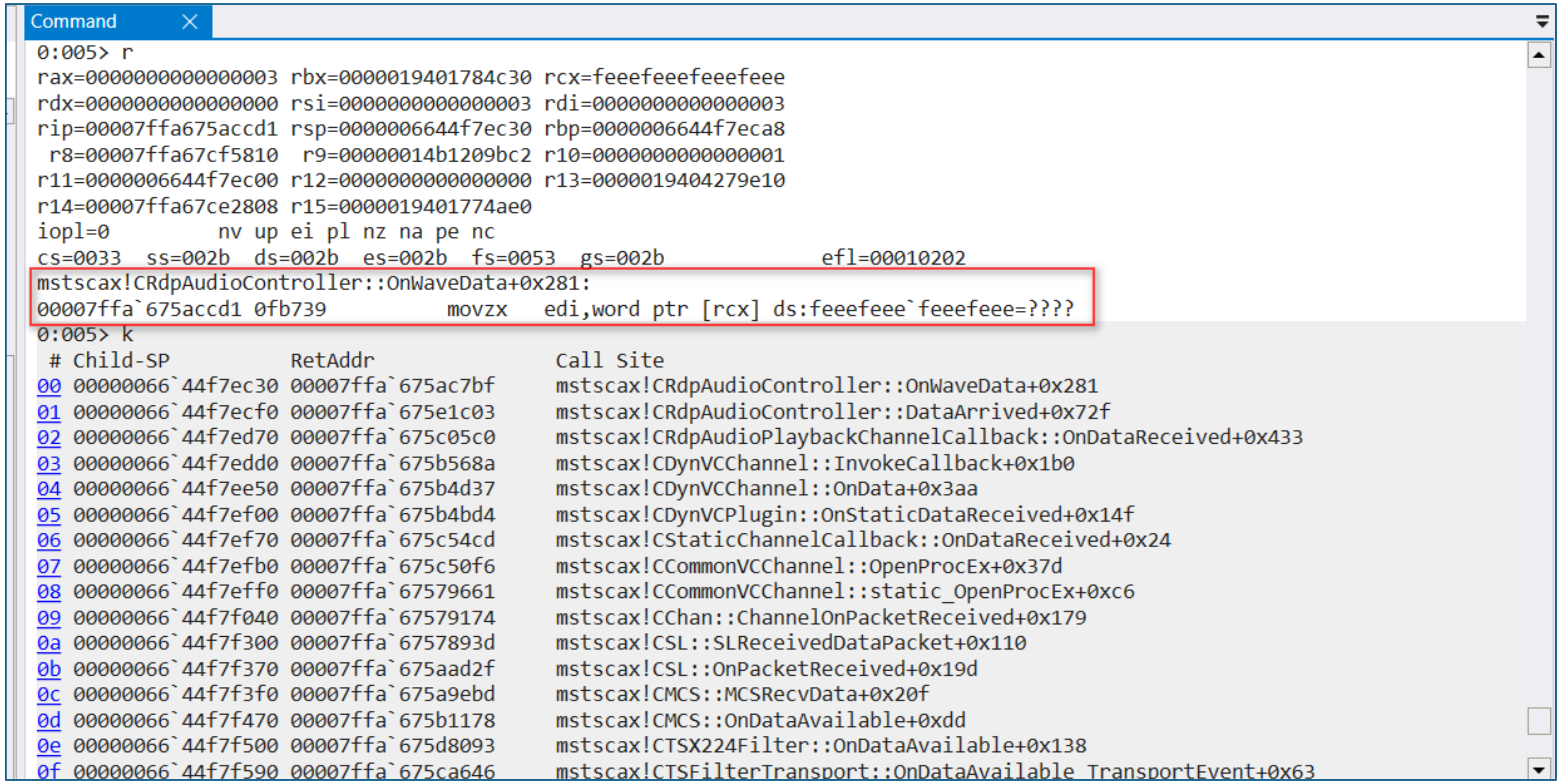
Remote Desktop Services enables users to access virtual desktops, session-based desktops, and RemoteApp programs. Use the Remote Desktop Services installation to configure a Virtual machine-based or a Session-based desktop deployment.

Windows Server

Start Fuzzing



Guideboard: An Old Unfixed OOBR

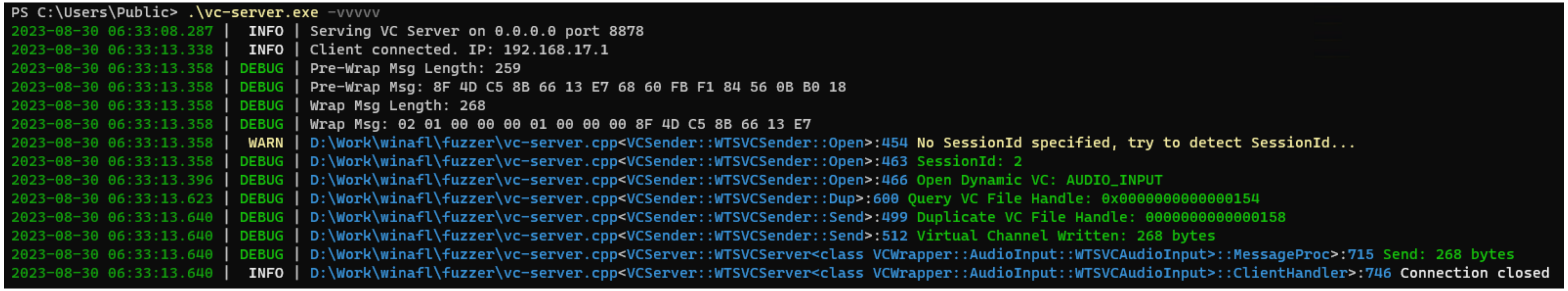


```
0:005> r
rax=0000000000000003 rbx=0000019401784c30 rcx=feeefeeefeeefeee
rdx=0000000000000000 rsi=0000000000000003 rdi=0000000000000003
rip=00007ffa675accd1 rsp=0000006644f7ec30 rbp=0000006644f7eca8
 r8=00007ffa67cf5810  r9=00000014b1209bc2 r10=0000000000000001
r11=0000006644f7ec00 r12=0000000000000000 r13=0000019404279e10
r14=00007ffa67ce2808 r15=0000019401774ae0
iopl=0         nv up ei pl nz na pe nc
cs=0033  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010202
mstscax!CRdpAudioController::OnWaveData+0x281:
00007ffa`675accd1 0fb739          movzx   edi,word ptr [rcx] ds:feeefeee`feeefeee=????
0:005> k
 # Child-SP          RetAddr               Call Site
00 00000066`44f7ec30 00007ffa`675ac7bf     mstscax!CRdpAudioController::OnWaveData+0x281
01 00000066`44f7ecf0 00007ffa`675e1c03     mstscax!CRdpAudioController::DataArrived+0x72f
02 00000066`44f7ed70 00007ffa`675c05c0     mstscax!CRdpAudioPlaybackChannelCallback::OnDataReceived+0x433
03 00000066`44f7edd0 00007ffa`675b568a     mstscax!CDynVCChannel::InvokeCallback+0x1b0
04 00000066`44f7ee50 00007ffa`675b4d37     mstscax!CDynVCChannel::OnData+0x3aa
05 00000066`44f7ef00 00007ffa`675b4bd4     mstscax!CDynVCPlugin::OnStaticDataReceived+0x14f
06 00000066`44f7ef70 00007ffa`675c54cd     mstscax!CStaticChannelCallback::OnDataReceived+0x24
07 00000066`44f7efb0 00007ffa`675c50f6     mstscax!CCommonVCChannel::OpenProcEx+0x37d
08 00000066`44f7eff0 00007ffa`67579661     mstscax!CCommonVCChannel::static_OpenProcEx+0xc6
09 00000066`44f7f040 00007ffa`67579174     mstscax!CChan::ChannelOnPacketReceived+0x179
0a 00000066`44f7f300 00007ffa`6757893d     mstscax!CSL::SLReceivedDataPacket+0x110
0b 00000066`44f7f370 00007ffa`675aad2f     mstscax!CSL::OnPacketReceived+0x19d
0c 00000066`44f7f3f0 00007ffa`675a9ebd     mstscax!CMCS::MCSRecvData+0x20f
0d 00000066`44f7f470 00007ffa`675b1178     mstscax!CMCS::OnDataAvailable+0xdd
0e 00000066`44f7f500 00007ffa`675d8093     mstscax!CTSX224Filter::OnDataAvailable+0x138
0f 00000066`44f7f590 00007ffa`675ca646     mstscax!CTSFilterTransport::OnDataAvailable_TransportEvent+0x63
```

Same bug with: https://blog.thalium.re/posts/fuzzing-microsoft-rdp-client-using-virtual-channels/#out-of-bounds-read-in-rdpsnd

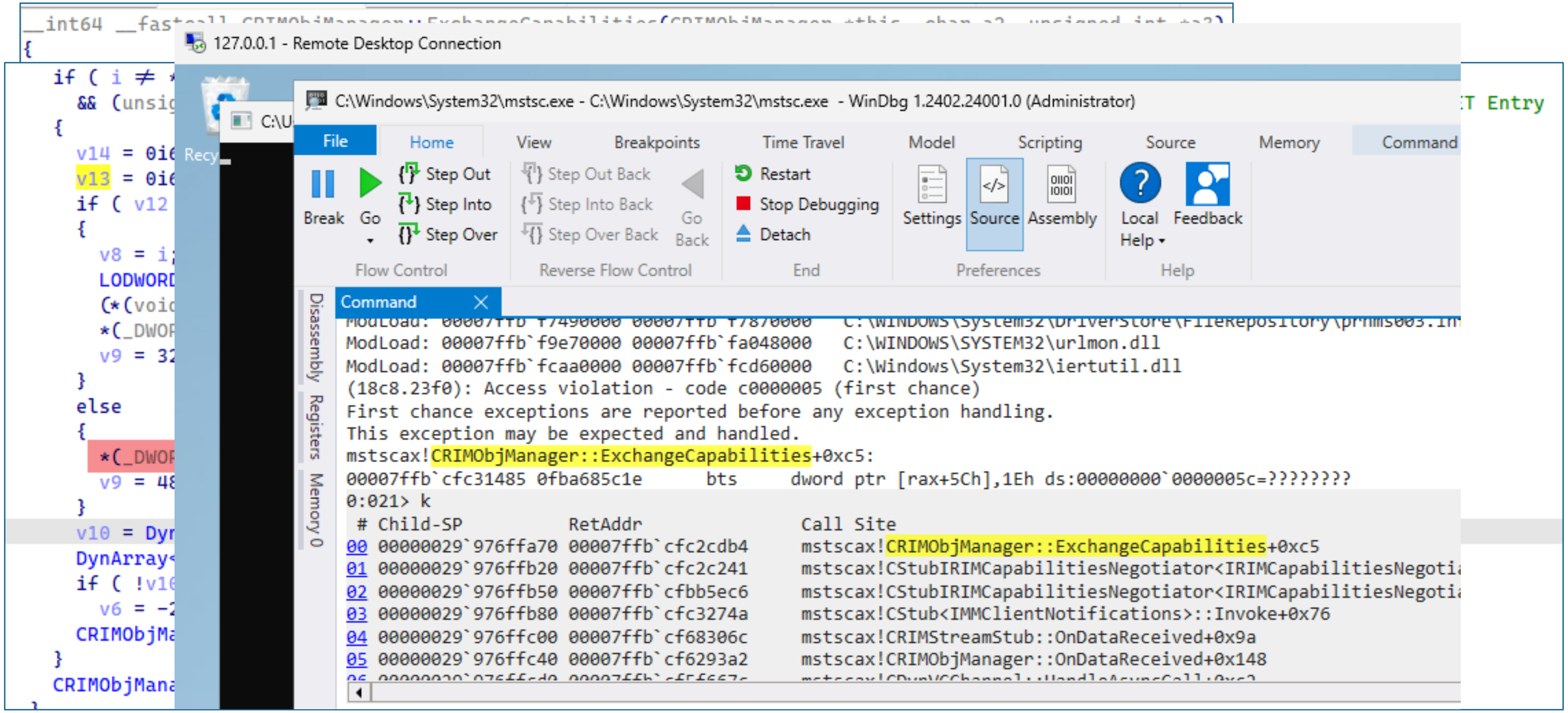
Enhancing Fuzzing

- **WinAFL**
  - Transplant the mutation strategy of honggfuzz
  - Coverage visualization & statistics
  - Fuzzer arch **#1** to **#2** (**Loop** -> **Proxy**)
- **Reversing**
- **RTFM**



```
PS C:\Users\Public> .\vc-server.exe -vvvvv
2023-08-30 06:33:08.287 |  INFO | Serving VC Server on 0.0.0.0 port 8878
2023-08-30 06:33:13.338 |  INFO | Client connected. IP: 192.168.17.1
2023-08-30 06:33:13.358 | DEBUG | Pre-Wrap Msg Length: 259
2023-08-30 06:33:13.358 | DEBUG | Pre-Wrap Msg: 8F 4D C5 8B 66 13 E7 68 60 FB F1 84 56 0B B0 18
2023-08-30 06:33:13.358 | DEBUG | Wrap Msg Length: 268
2023-08-30 06:33:13.358 | DEBUG | Wrap Msg: 02 01 00 00 00 01 00 00 00 8F 4D C5 8B 66 13 E7
2023-08-30 06:33:13.358 |  WARN | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Open>:454 No SessionId specified, try to detect SessionId...
2023-08-30 06:33:13.358 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Open>:463 SessionId: 2
2023-08-30 06:33:13.396 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Open>:466 Open Dynamic VC: AUDIO_INPUT
2023-08-30 06:33:13.623 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Dup>:600 Query VC File Handle: 0x0000000000000154
2023-08-30 06:33:13.640 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Send>:499 Duplicate VC File Handle: 0000000000000158
2023-08-30 06:33:13.640 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCSender::WTSVCSender::Send>:512 Virtual Channel Written: 268 bytes
2023-08-30 06:33:13.640 | DEBUG | D:\Work\winafl\fuzzer\vc-server.cpp<VCServer::WTSVCServer<class VCWrapper::AudioInput::WTSVCAudioInput>::MessageProc>:715 Send: 268 bytes
2023-08-30 06:33:13.640 |  INFO | D:\Work\winafl\fuzzer\vc-server.cpp<VCServer::WTSVCServer<class VCWrapper::AudioInput::WTSVCAudioInput>::ClientHandler>:746 Connection closed
```

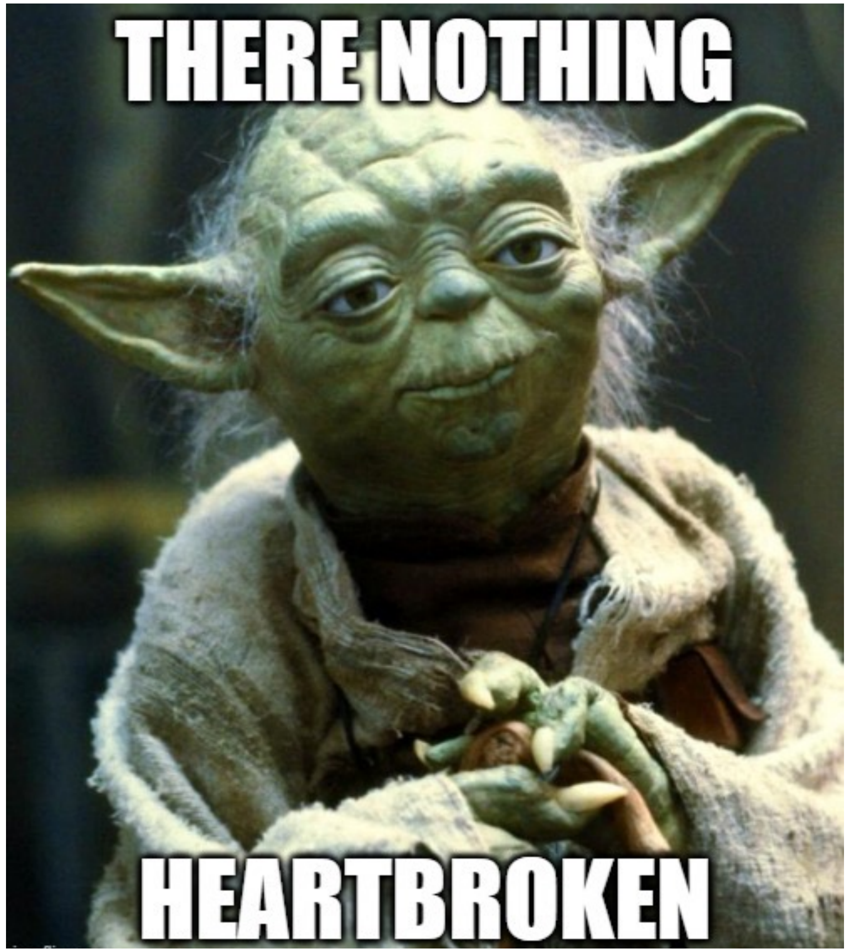
Dream Start: A New NPD (Won’t Fix)



```
__int64 __fastcall CRIMObjManager::ExchangeCapabilities(CRIMObjManager *this, char a2, unsigned int *a3)
{
  if ( i ≠ *
    && (unsig
  {
    v14 = 0i6
    v13 = 0i6
    if ( v12
    {
      v8 = i;
      LODWORD
      (*(voic
      *(_DWOR
      v9 = 32
    }
    else
    {
      *(_DWOR
      v9 = 48
    }
    v10 = Dyr
    DynArray<
    if ( !v10
      v6 = -2
    CRIMObjMa
  }
  CRIMObjMana
}
```

```
ModLoad: 00007ffb`f7490000 00007ffb`f7870000   C:\WINDOWS\System32\DriverStore\FileRepository\prnms003.in
ModLoad: 00007ffb`f9e70000 00007ffb`fa048000   C:\WINDOWS\SYSTEM32\urlmon.dll
ModLoad: 00007ffb`fcaa0000 00007ffb`fcd60000   C:\Windows\System32\iertutil.dll
(18c8.23f0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
mstscax!CRIMObjManager::ExchangeCapabilities+0xc5:
00007ffb`cfc31485 0fba685c1e      bts     dword ptr [rax+5Ch],1Eh ds:00000000`0000005c=????????
0:021> k
 # Child-SP          RetAddr               Call Site
00 00000029`976ffa70 00007ffb`cfc2cdb4     mstscax!CRIMObjManager::ExchangeCapabilities+0xc5
01 00000029`976ffb20 00007ffb`cfc2c241     mstscax!CStubIRIMCapabilitiesNegotiator<IRIMCapabilitiesNegoti
02 00000029`976ffb50 00007ffb`cfbb5ec6     mstscax!CStubIRIMCapabilitiesNegotiator<IRIMCapabilitiesNegoti
03 00000029`976ffb80 00007ffb`cfc3274a     mstscax!CStub<IMMClientNotifications>::Invoke+0x76
04 00000029`976ffc00 00007ffb`cf68306c     mstscax!CRIMStreamStub::OnDataReceived+0x9a
05 00000029`976ffc40 00007ffb`cf6293a2     mstscax!CRIMObjManager::OnDataReceived+0x148
06 00000029`976ffcd0 00007ffb`cf5f667c     mstscax!CDynVCChannel::HandleAsyncCall+0xc2
```

Check & Doubt



Eureka: Race Condition

black hat USA 2023

Race Condition – What the Developer Think

black hat USA 2023

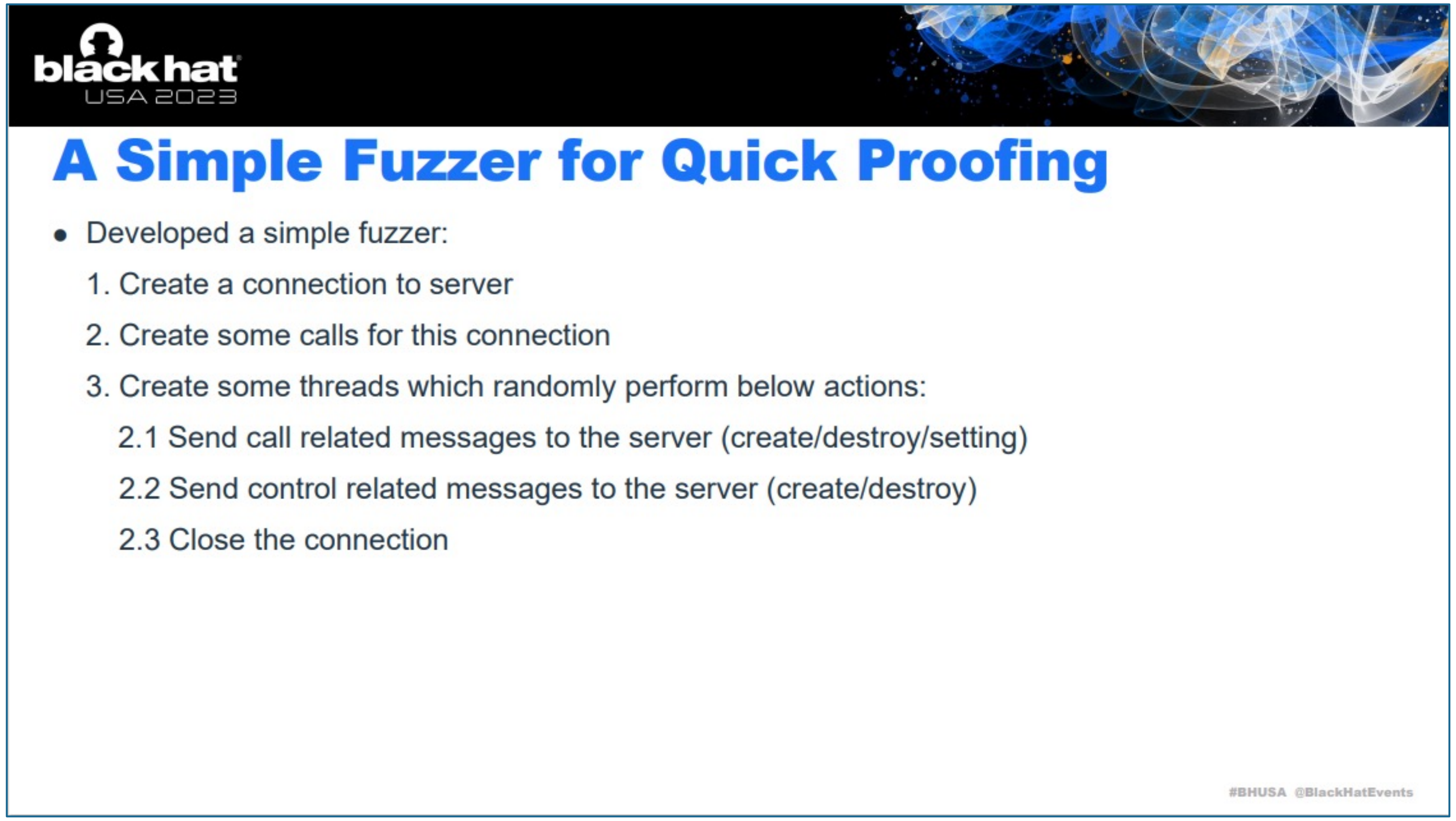
Race Condition – What’s Actual Happening

#BHUSA @BlackHatEvents

https://i.blackhat.com/BH-US-23/Presentations/US-23-YukiChen-Diving-into-Windows-Remote-Access.pdf

New Fuzzer

- Developed a simple Fuzzer



black hat USA 2023

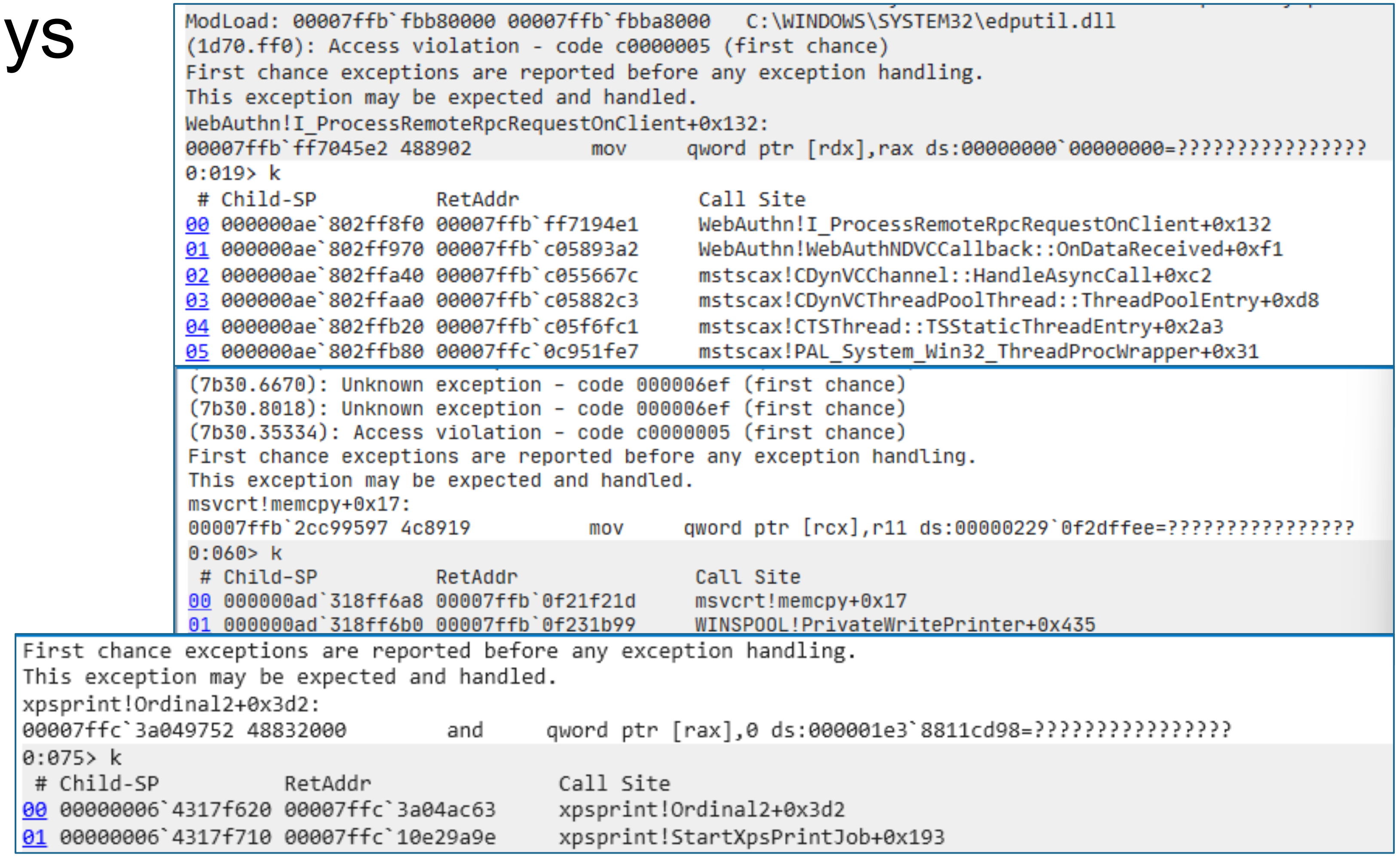
A Simple Fuzzer for Quick Proofing

- Developed a simple fuzzer:
  1. Create a connection to server
  2. Create some calls for this connection
  3. Create some threads which randomly perform below actions:
     2.1 Send call related messages to the server (create/destroy/setting)
     2.2 Send control related messages to the server (create/destroy)
     2.3 Close the connection

#BHUSA @BlackHatEvents

New World

- Got a few crashes in days
- Manual auditing



```
ModLoad: 00007ffb`fbb80000 00007ffb`fbba8000   C:\WINDOWS\SYSTEM32\edputil.dll
(1d70.ff0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
WebAuthn!I_ProcessRemoteRpcRequestOnClient+0x132:
00007ffb`ff7045e2 488902          mov     qword ptr [rdx],rax ds:00000000`00000000=????????????????
0:019> k
 # Child-SP          RetAddr               Call Site
00 000000ae`802ff8f0 00007ffb`ff7194e1     WebAuthn!I_ProcessRemoteRpcRequestOnClient+0x132
01 000000ae`802ff970 00007ffb`c05893a2     WebAuthn!WebAuthNDVCCallback::OnDataReceived+0xf1
02 000000ae`802ffa40 00007ffb`c055667c     mstscax!CDynVCChannel::HandleAsyncCall+0xc2
03 000000ae`802ffaa0 00007ffb`c05882c3     mstscax!CDynVCThreadPoolThread::ThreadPoolEntry+0xd8
04 000000ae`802ffb20 00007ffb`c05f6fc1     mstscax!CTSThread::TSStaticThreadEntry+0x2a3
05 000000ae`802ffb80 00007ffc`0c951fe7     mstscax!PAL_System_Win32_ThreadProcWrapper+0x31
```

```
(7b30.6670): Unknown exception - code 000006ef (first chance)
(7b30.8018): Unknown exception - code 000006ef (first chance)
(7b30.35334): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
msvcrt!memcpy+0x17:
00007ffb`2cc99597 4c8919          mov     qword ptr [rcx],r11 ds:00000229`0f2dffee=????????????????
0:060> k
 # Child-SP          RetAddr               Call Site
00 000000ad`318ff6a8 00007ffb`0f21f21d     msvcrt!memcpy+0x17
01 000000ad`318ff6b0 00007ffb`0f231b99     WINSPOOL!PrivateWritePrinter+0x435
```

```
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
xpsprint!Ordinal2+0x3d2:
00007ffc`3a049752 48832000        and     qword ptr [rax],0 ds:000001e3`8811cd98=????????????????
0:075> k
 # Child-SP          RetAddr               Call Site
00 00000006`4317f620 00007ffc`3a04ac63     xpsprint!Ordinal2+0x3d2
01 00000006`4317f710 00007ffc`10e29a9e     xpsprint!StartXpsPrintJob+0x193
```

Case Study

Case 01 - Normal Printer UAF

```
0:060> k
# Child-SP          RetAddr               Call Site
00 000000ad`318ff6a8 00007ffb`0f21f21d     msvcrt!memcpy+0x17
01 000000ad`318ff6b0 00007ffb`0f231b99     WINSPOOL!PrivateWritePrinter+0x435
02 000000ad`318ffbe0 00007ffa`4c1c8c40     WINSPOOL!WritePrinter+0x9
03 000000ad`318ffc20 00007ffa`4c1c1fea     mstscax!W32DrAutoPrn::AsyncWriteIOFunc+0x3d0

……

WINSPOOL!Ordinal361+0x182:
00007ffc`5080a942 83bfb000000002   cmp     dword ptr [rdi+0B0h],2 ds:0000024d`1e422fa0=????????
0:029> k
 # Child-SP          RetAddr               Call Site
00 00000063`ea1ffa70 00007ffc`507fe72b     WINSPOOL!Ordinal361+0x182
01 00000063`ea1ffab0 00007ffc`5080d6e4     WINSPOOL!StartDocDlgW+0x67b
02 00000063`ea1ffdb0 00007ffc`10d89770     WINSPOOL!StartDocPrinterW+0xe4
03 00000063`ea1ffe00 00007ffc`10d82cea     mstscax!W32DrAutoPrn::AsyncWriteIOFunc+0x200
```

Case 01 - Normal Printer UAF

Thread 1 – Worker thread

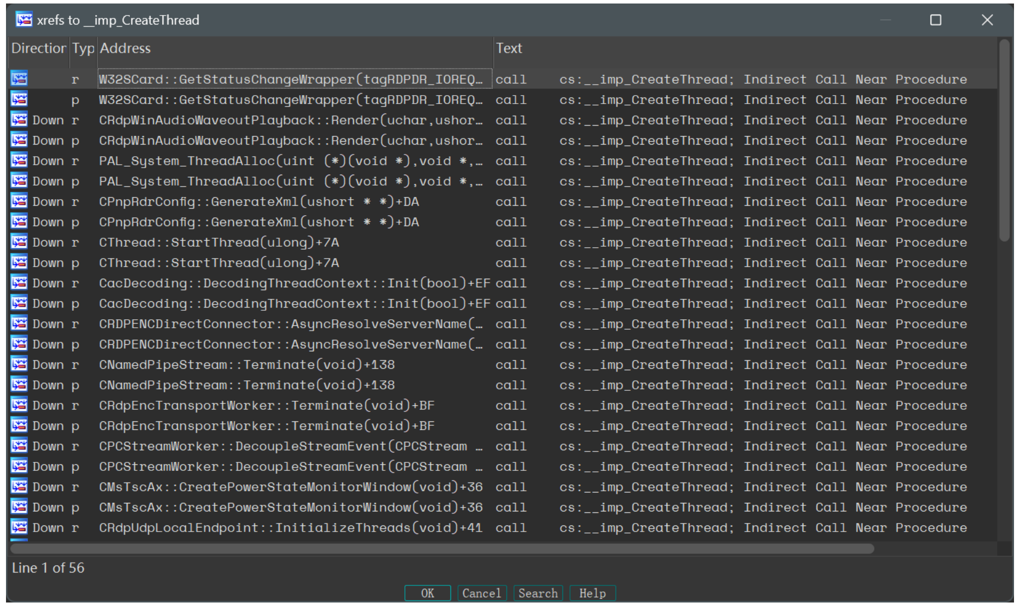
```
W32DrAutoPrn::AsyncWriteIOFunc
{
  // ...

  if (bUseXpsMode) CALL W32DrAutoPrn::StartXPSJob;

  CALL OpenPrinterW; // 1. Get the printer handle

  // ... Race window ...

  CALL WritePrinter; // 3. Use the printer handle

  // ...

}
```

Thread 2 – Close Printer Thread

```
W32DrAutoPrn::ClosePrinter
{
  // ...

  if (bUseXpsMode) CALL W32DrAutoPrn::CloseXPSJob;

  CALL EndPagePrinter;

  CALL EndDocPrinter;

  CALL ClosePrinter; // 2. Free the printer handle

  // ...

}
```

Case 02 - XPS Printer UAF

- Are there any other points?
- **CreateThread()** function
- Free and Use

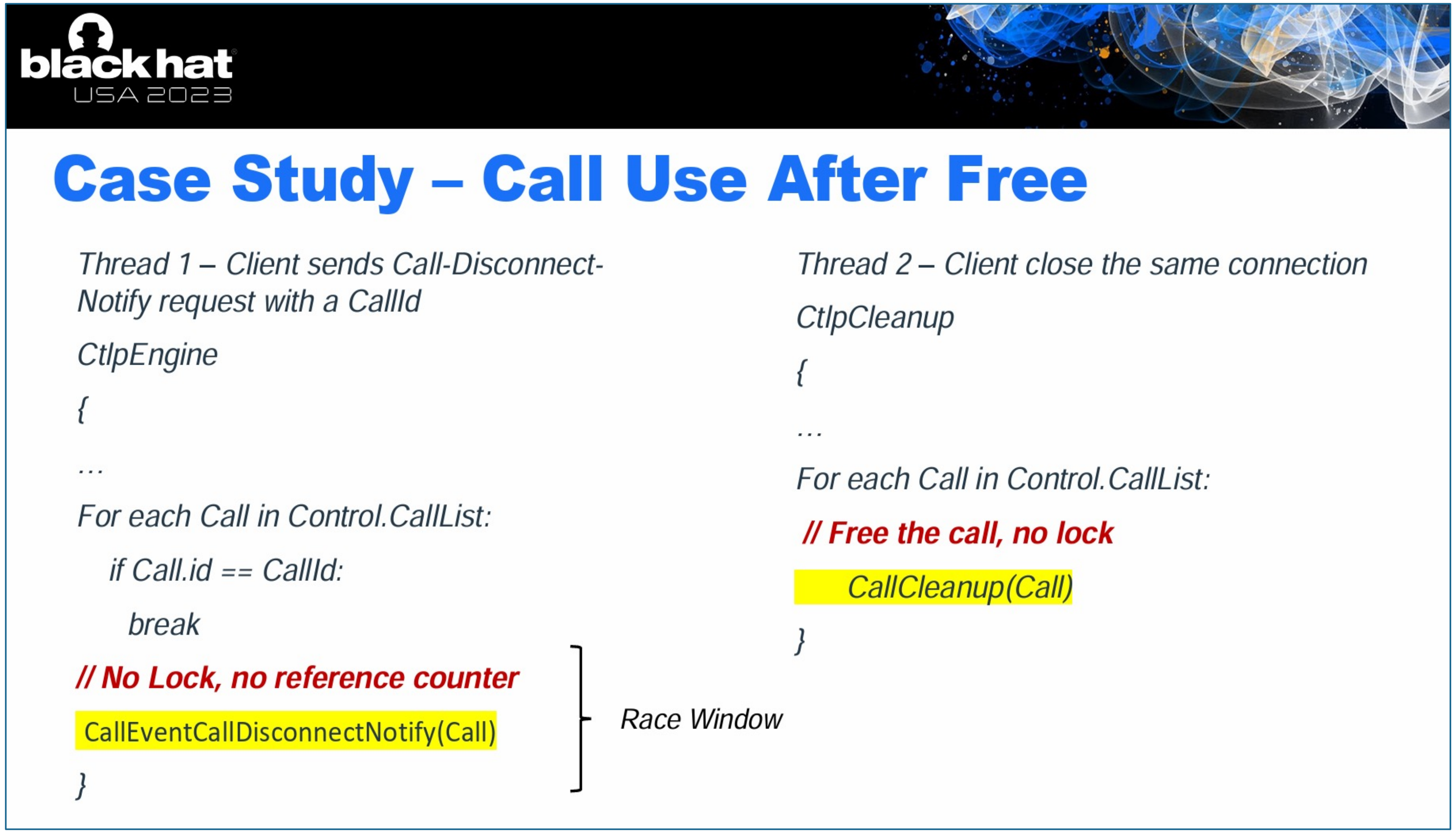


xrefs to __imp_CreateThread

Direction	Typ	Address	Text
	r	W32SCard::GetStatusChangeWrapper(tagRDPDR_IOREQ…	call cs:__imp_CreateThread; Indirect Call Near Procedure
	p	W32SCard::GetStatusChangeWrapper(tagRDPDR_IOREQ…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CRdpWinAudioWaveoutPlayback::Render(uchar,ushor…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CRdpWinAudioWaveoutPlayback::Render(uchar,ushor…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	PAL_System_ThreadAlloc(uint (*)(void *),void *,…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	PAL_System_ThreadAlloc(uint (*)(void *),void *,…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CPnpRdrConfig::GenerateXml(ushort * *)+DA	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CPnpRdrConfig::GenerateXml(ushort * *)+DA	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CThread::StartThread(ulong)+7A	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CThread::StartThread(ulong)+7A	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CacDecoding::DecodingThreadContext::Init(bool)+EF	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CacDecoding::DecodingThreadContext::Init(bool)+EF	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CRDPENCDirectConnector::AsyncResolveServerName(…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CRDPENCDirectConnector::AsyncResolveServerName(…	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CNamedPipeStream::Terminate(void)+138	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CNamedPipeStream::Terminate(void)+138	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CRdpEncTransportWorker::Terminate(void)+BF	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CRdpEncTransportWorker::Terminate(void)+BF	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CPCStreamWorker::DecoupleStreamEvent(CPCStream …	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CPCStreamWorker::DecoupleStreamEvent(CPCStream …	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CMsTscAx::CreatePowerStateMonitorWindow(void)+36	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	p	CMsTscAx::CreatePowerStateMonitorWindow(void)+36	call cs:__imp_CreateThread; Indirect Call Near Procedure
Down	r	CRdpUdpLocalEndpoint::InitializeThreads(void)+41	call cs:__imp_CreateThread; Indirect Call Near Procedure

Line 1 of 56

OK
Cancel
Search
Help



black hat USA 2023

Case Study – Call Use After Free

Thread 1 – Client sends Call-Disconnect-Notify request with a CallId

```
CtlpEngine
{
…
For each Call in Control.CallList:
    if Call.id == CallId:
        break
// No Lock, no reference counter
CallEventCallDisconnectNotify(Call)
}
```

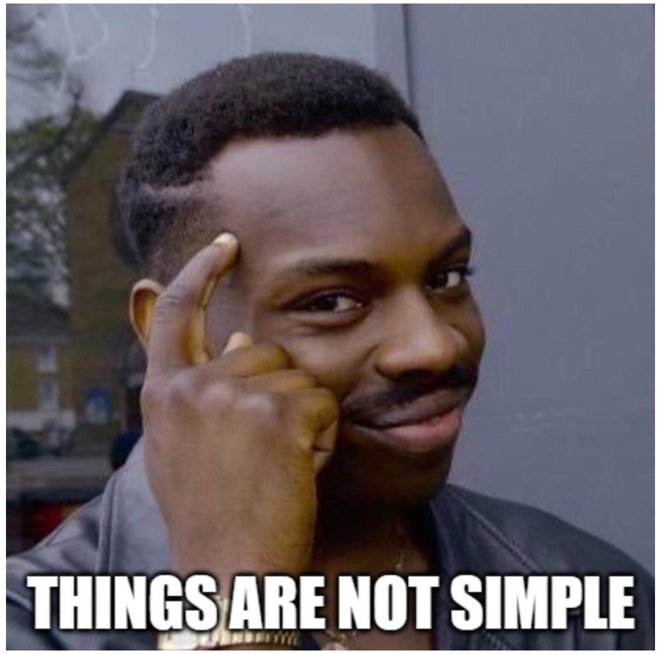
Race Window

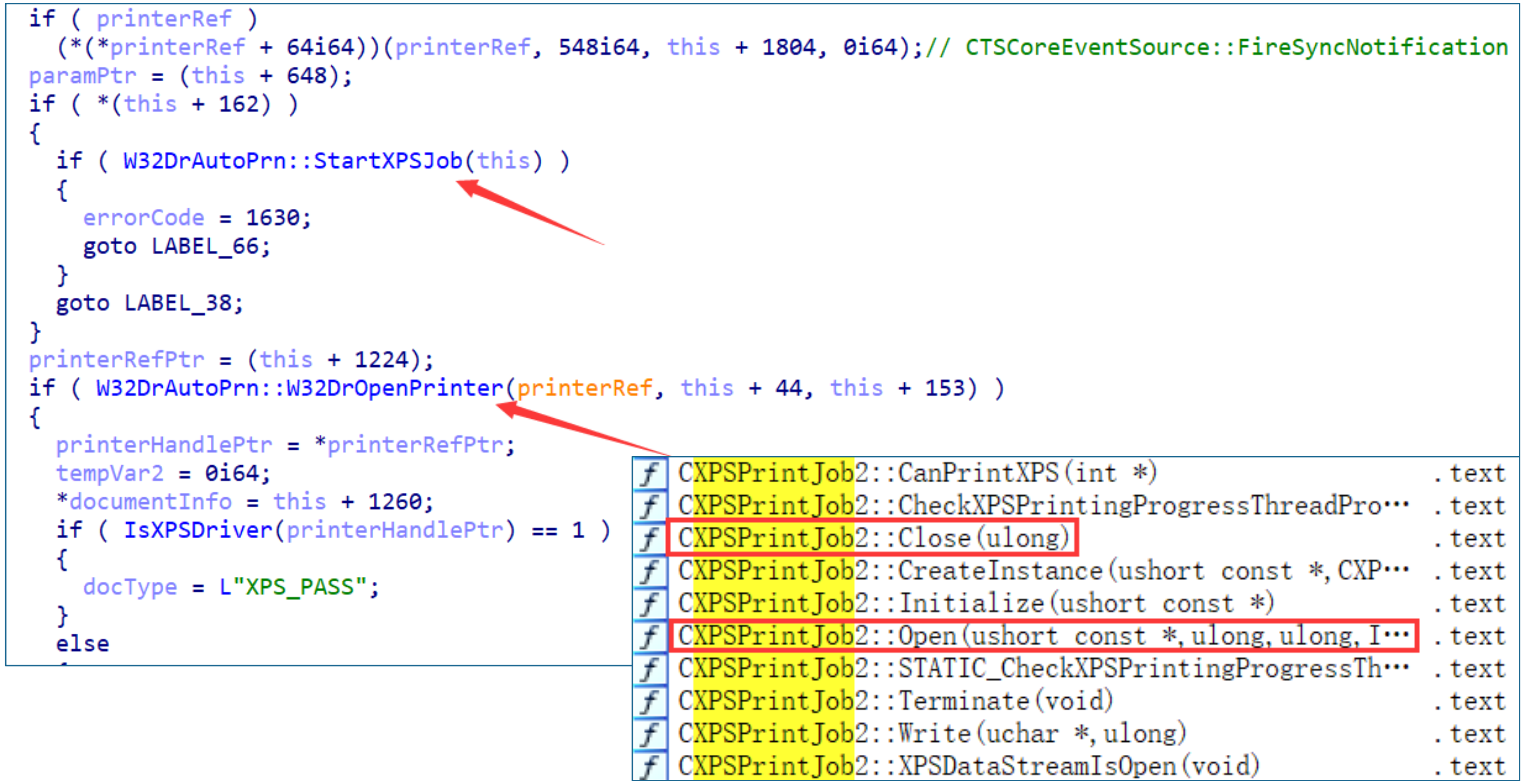
Thread 2 – Client close the same connection

```
CtlpCleanup
{
…
For each Call in Control.CallList:
// Free the call, no lock
    CallCleanup(Call)
}
```

Case 02 - XPS Printer UAF

- Variant analysis
- Targeted test

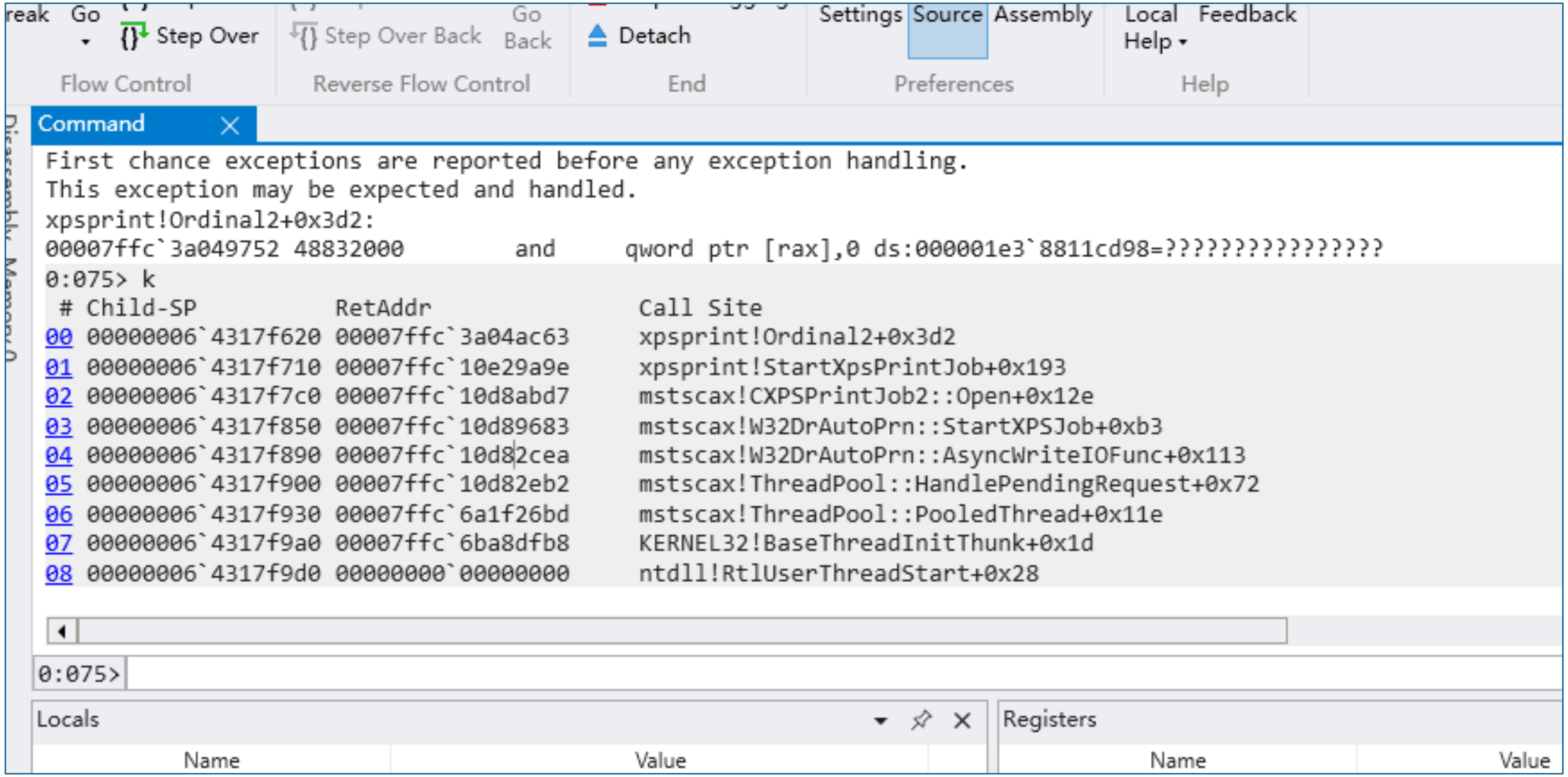




```
if ( printerRef )
  (*(*printerRef + 64i64))(printerRef, 548i64, this + 1804, 0i64);// CTSCoreEventSource::FireSyncNotification
paramPtr = (this + 648);
if ( *(this + 162) )
{
  if ( W32DrAutoPrn::StartXPSJob(this) )
  {
    errorCode = 1630;
    goto LABEL_66;
  }
  goto LABEL_38;
}
printerRefPtr = (this + 1224);
if ( W32DrAutoPrn::W32DrOpenPrinter(printerRef, this + 44, this + 153) )
{
  printerHandlePtr = *printerRefPtr;
  tempVar2 = 0i64;
  *documentInfo = this + 1260;
  if ( IsXPSDriver(printerHandlePtr) == 1 )
  {
    docType = L"XPS_PASS";
  }
  else
```

```
f CXPSPrintJob2::CanPrintXPS(int *)                              .text
f CXPSPrintJob2::CheckXPSPrintingProgressThreadPro…              .text
f CXPSPrintJob2::Close(ulong)                                    .text
f CXPSPrintJob2::CreateInstance(ushort const *,CXP…              .text
f CXPSPrintJob2::Initialize(ushort const *)                      .text
f CXPSPrintJob2::Open(ushort const *,ulong,ulong,I…              .text
f CXPSPrintJob2::STATIC_CheckXPSPrintingProgressTh…              .text
f CXPSPrintJob2::Terminate(void)                                 .text
f CXPSPrintJob2::Write(uchar *,ulong)                            .text
f CXPSPrintJob2::XPSDataStreamIsOpen(void)                       .text
```

Case 02 - XPS Printer UAF



```
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
xpsprint!Ordinal2+0x3d2:
00007ffc`3a049752 48832000        and     qword ptr [rax],0 ds:000001e3`8811cd98=????????????????
0:075> k
 # Child-SP          RetAddr               Call Site
00 00000006`4317f620 00007ffc`3a04ac63     xpsprint!Ordinal2+0x3d2
01 00000006`4317f710 00007ffc`10e29a9e     xpsprint!StartXpsPrintJob+0x193
02 00000006`4317f7c0 00007ffc`10d8abd7     mstscax!CXPSPrintJob2::Open+0x12e
03 00000006`4317f850 00007ffc`10d89683     mstscax!W32DrAutoPrn::StartXPSJob+0xb3
04 00000006`4317f890 00007ffc`10d82cea     mstscax!W32DrAutoPrn::AsyncWriteIOFunc+0x113
05 00000006`4317f900 00007ffc`10d82eb2     mstscax!ThreadPool::HandlePendingRequest+0x72
06 00000006`4317f930 00007ffc`6a1f26bd     mstscax!ThreadPool::PooledThread+0x11e
07 00000006`4317f9a0 00007ffc`6ba8dfb8     KERNEL32!BaseThreadInitThunk+0x1d
08 00000006`4317f9d0 00000000`00000000     ntdll!RtlUserThreadStart+0x28

0:075>
```

Case 02 - XPS Printer UAF

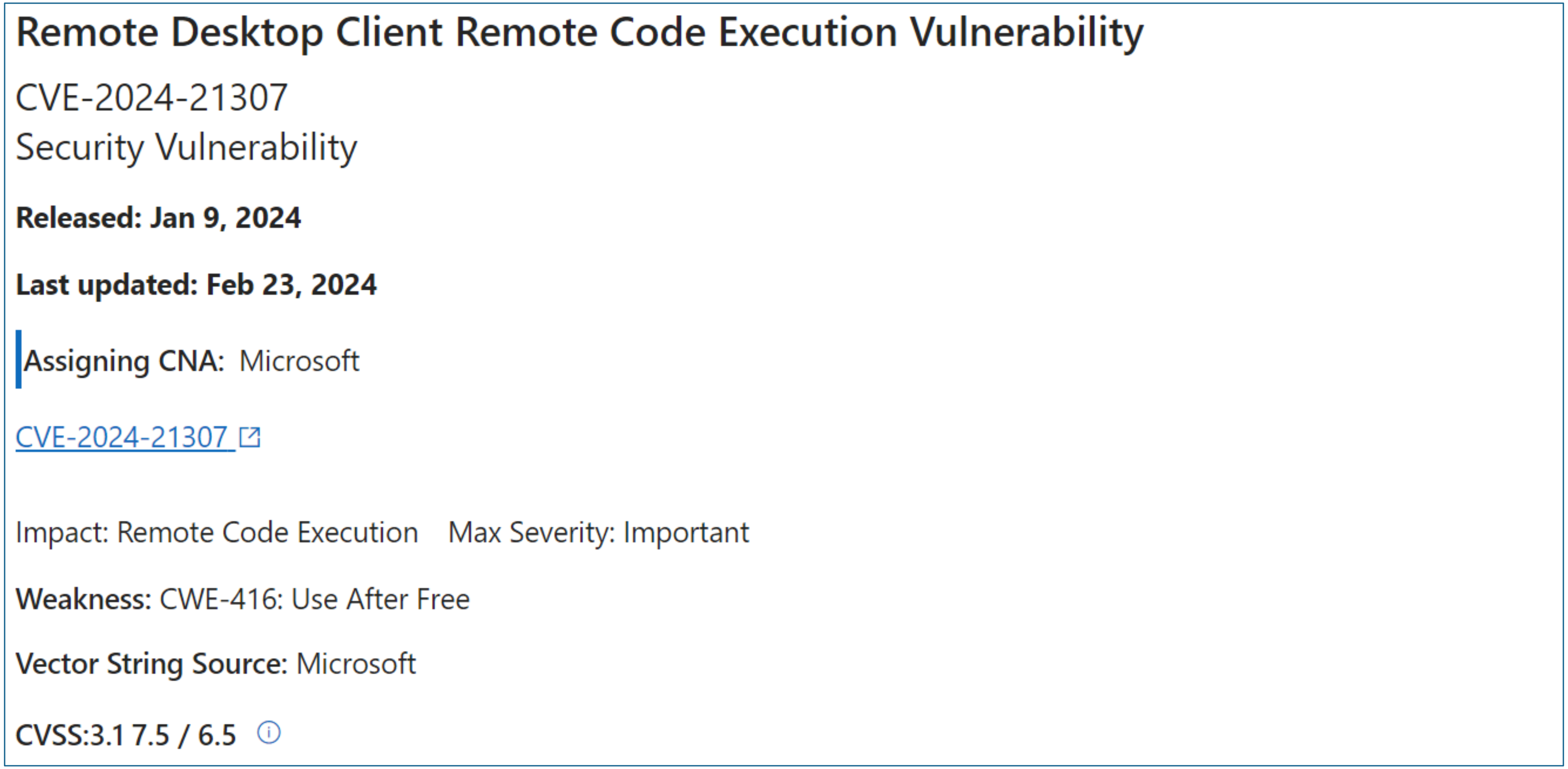
Thread 1 – Send Creat PDU To Load xpsprint.dll

```
W32DrAutoPrn::StartXPSJob()
{
 CXPSPrintJob2::Initialize
 {
  // Load xpsprint.dll
  library = LoadLibraryExW(L"xpsprint.dll",0,0x800u);
 }

 CXPSPrintJob2::Open(pXPSJob)
 {
  if (CXPSPrintJob2::XPSDataStreamIsOpen(this) )
  {
   return 0x8007139;
  }

  // ... Race window ...

  // Use some pointer in xpsprint.dll and crash !
  TempFile = StartXpsPrintJob();
 }
}
```

Thread 2 – Send Close PDU To Free xpsprint.dll

```
CXPSPrintJob2::Close()
{
 if ( !CXPSPrintJob2::XPSDataStreamIsOpen(this) )
 {
   return 0x8007139;
 }
 CXPSPrintJob2::~CXPSPrintJob2
 {
  CXPSPrintJob2::Terminate(pXPSJob)
  {
   // Unload xpsprint.dll !
   FreeLibrary(xpsprint.dll);
  }
 }
}
```

Patches



Remote Desktop Client Remote Code Execution Vulnerability

CVE-2024-21307
Security Vulnerability

Released: Jan 9, 2024

Last updated: Feb 23, 2024

Assigning CNA: Microsoft

CVE-2024-21307

Impact: Remote Code Execution   Max Severity: Important

Weakness: CWE-416: Use After Free

Vector String Source: Microsoft

CVSS:3.1 7.5 / 6.5

https://msrc.microsoft.com/update-guide/vulnerability/CVE-2024-21307

Patches

Patches – CVE-2024-21307 #1

```
W32DrAutoPrn::StartXPSJob
{

  +CALL EnterCriticalSection;

  // ...

  CALL Create_CXPSPrintJob(&_ptrXPSJob, ...);

  CALL _ptrXPSJob->Open(_ptrXPSJob, ...);

  // ...

  +CALL LeaveCriticalSection;

}
```

Patches – CVE-2024-21307 #2

```
W32DrAutoPrn::CloseXPSJob
{

  // ...

  +CALL EnterCriticalSection;

  // ...

  +CALL LeaveCriticalSection;

  // ...


}
```

Future

Future Work

RDP Server

More Channels

More Protocols

…

Black Hat Sound Bytes

- We have shared some skills on fuzzing Windows RDP components
- We have shared our latest research on Windows RDP Client vulnerability
- We have showed the significance of race condition in vulnerability discovery

Thanks!

References

1. https://github.com/cyberark/RDPFuzz
2. https://github.com/Team-BT5/WinAFL-RDP
3. https://blog.thalium.re/posts/misc/rdpegfx/Hexacon2022-Fuzzing_RDPEGFX_with_wtf.pdf
4. https://i.blackhat.com/BH-US-23/Presentations/US-23-YukiChen-Diving-into-Windows-Remote-Access.pdf
5. https://i.blackhat.com/eu-19/Wednesday/eu-19-Park-Fuzzing-And-Exploiting-Virtual-Channels-In-Microsoft-Remote-Desktop-Protocol-For-Fun-And-Profit-4.pdf
6. https://www.sstic.org/media/SSTIC2022/SSTIC-actes/fuzzing_microsofts_rdp_client_using_virtual_channe/SSTIC2022-Article-fuzzing_microsofts_rdp_client_using_virtual_channels-ricotta.pdf
7. https://conference.hitb.org/hitbsecconf2021sin/materials/D2T1%20-%20Holding%20The%20Stick%20at%20Both%20Ends%20-%20Fuzzing%20RDP%20Client%20and%20Server%20-%20Shaked%20Reiner%20&%20Or%20Ben-Porath.pdf