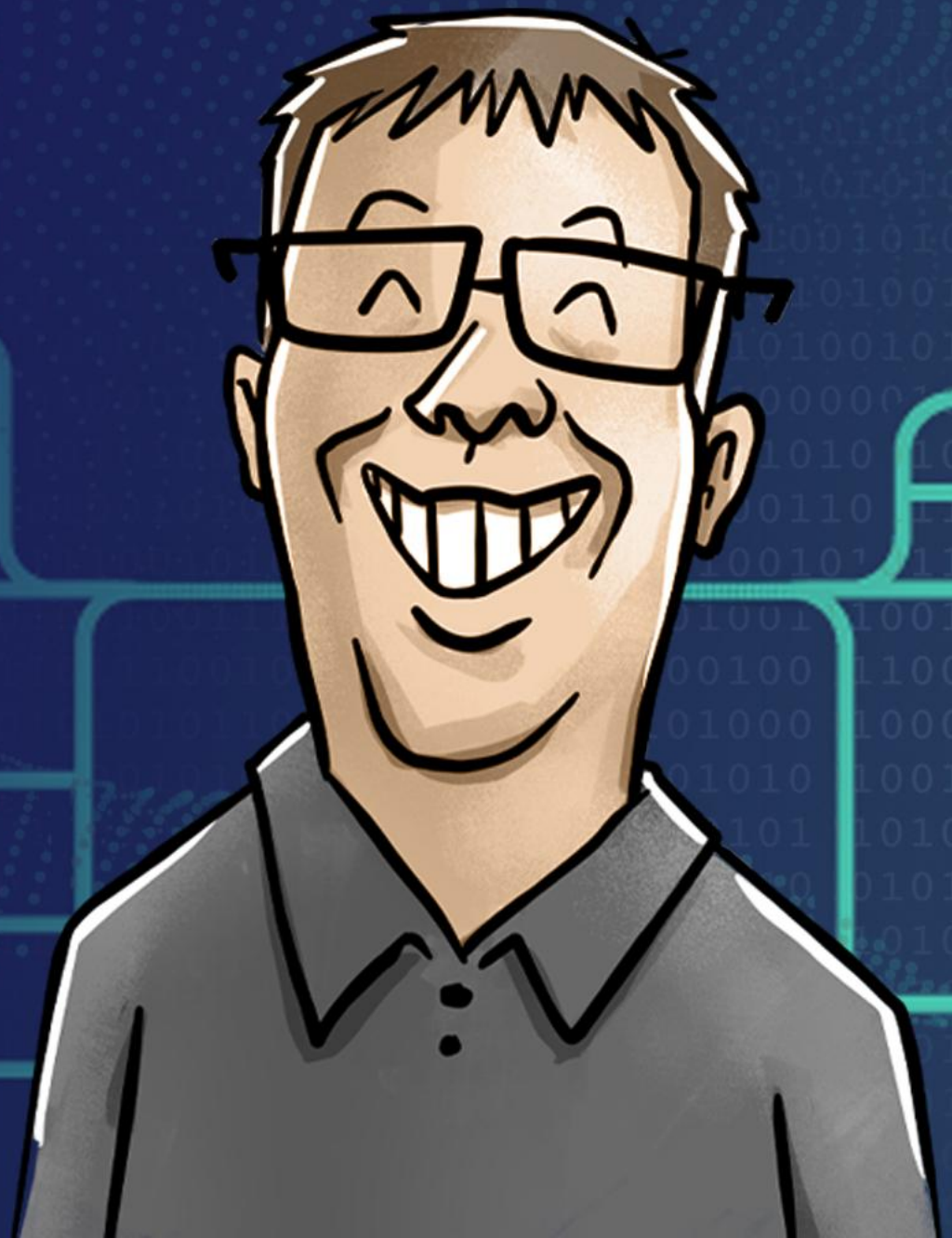
Devolutions

Decrypting RDP traffic in Wireshark

Marc-André Moreau
CTO, Devolutions



TLS pre-master secret logging 🙂

The technique we'll be using - the best by far

TLS pre-master secrets are used to derive **session keys**

- The server private key is only used to authenticate the **key exchange**
  - With Perfect Forward Secrecy (PFS) cipher suites, which are now the norm
- SChannel in Windows isolates TLS pre-master secrets in LSA
  - **LSA is a protected process**, but all secrets are conveniently in one place
- Other TLS stacks don't isolate TLS pre-master secrets in LSA
  - Many applications support the "**SSLKEYLOGFILE**" environment variable

LSA secret extraction method references

lsass API hooking blog post reference

https://b.poc.fun/decrypting-schannel-tls-part-1/

PSDetour module (Detour in PowerShell)

https://github.com/jborean93/PSDetour

Original tls-keylogger.ps1 script from Jordan Borean

https://gist.github.com/jborean93/6c1f1b3130f2675f1618da56633eb1fa

Application-level packet logging 😐

Your application sees the bytes, but can you read them?

Why not simply dump the bytes from your application?

- Manual inspection of hex dumps is a time-consuming task

It's tricky to "cook" a Wireshark capture file properly

- You need to reconstruct TCP/IP headers with a fake client/server, etc.
- Wireshark dissectors get confused by missing TLS over TCP/3389, etc.

Network MITM Proxy ☹️

Houston, we have a TLS token binding problem

CredSSP has built-in MITM protection

- CredSSP server public key echo step
- NTLM/Kerberos channel binding token (CBT)
  - Also known as Extended Protection for Authentication (EPA)

RDP has pre-TLS traffic (X.224 negotiation)

- All the MITM and reverse proxying tools expect “clean” TLS

Server RSA private key ☹️

The best TLS 1.1 had to offer. Move on, forget about it.

It used to be a thing – please don’t try to force TLS 1.1 or TLS 1.2 with Perfect Forward Secrecy (PFS) ciphers disabled, you’ll only end up breaking Windows Update instead.

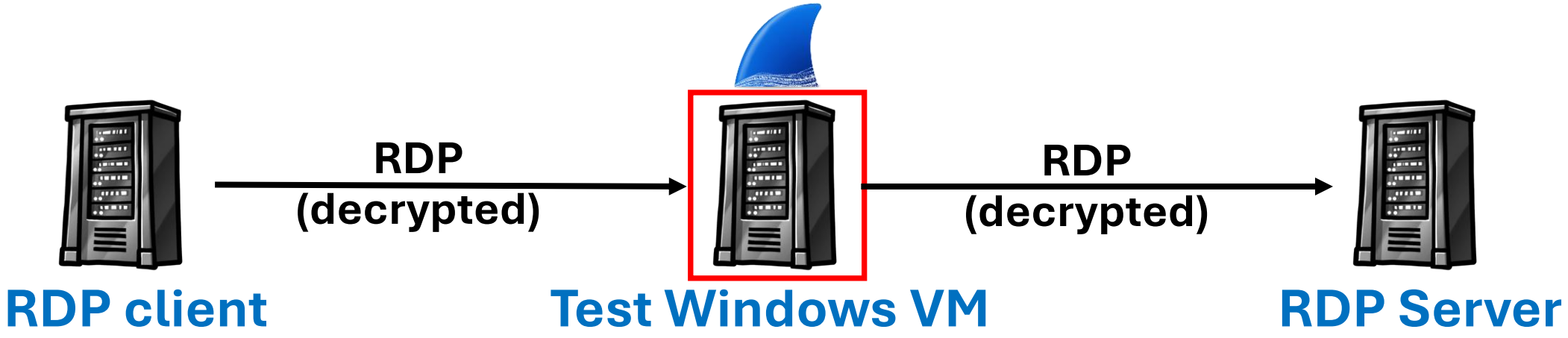
Summary of TLS decryption techniques

Technique	Description	Recommendation
TLS Pre-Master Secret Logging	Log TLS pre-master secrets into a text file (SSLKEYLOGFILE)	🙂 Yes!
Application-level packet logging	Log decrypted packet bytes from the TLS client application	😐 Maybe, in some cases
Network MITM proxy	Intercept, decrypt and re-encrypt TLS traffic using a different certificate	☹️ Not really an option
Server RSA Private Key	Use server RSA private key to decrypt corresponding TLS traffic	☹️ No longer viable

Devolutions

GETTING STARTED
with Live RDP decryption in Wireshark

Getting Started - Prerequisites



- LSA extended protection needs to be **disabled**
  - Don’t use a production device!
- RDP connections **to/from** the test VM can be decrypted
- Install Wireshark in the test VM for now (not your host)

Hyper-V Lab (Optional)

https://github.com/Devolutions/devolutions-labs

IT-HELP-RTR: Alpine Linux router with DHCP, NAT with host, etc.

IT-HELP-DC: domain controller with AD CS, root CA, HTTP CRL, etc.

IT-HELP-GW: RD Gateway, RDWeb, connection broker, licensing server

IT-HELP-WAC: Windows Admin Center

Virtual machine bootstrapping

Disable LSA extended protection, then reboot:

```
Set-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\Control\Lsa'
-Name 'RunAsPPL' -Value 0
```

Install PowerShell 7 (yes, it's required):

```
iex "& { $(irm https://aka.ms/install-powershell.ps1) } -UseMSI –Quiet"
```

Set the PowerShell execution policy to Unrestricted:

```
Set-ExecutionPolicy Unrestricted -Scope LocalMachine
```

Logging TLS secrets from LSA (SChannel)

Launch PowerShell 7 elevated, then install PSDetour:

Install-Module -Name PSDetour -Scope AllUsers –Force

Install the AwakeCoding.DebugTools PowerShell module:

Install-Module –Name AwakeCoding.DebugTools –Scope AllUsers –Force

Start logging TLS pre-master secrets, and leave terminal open:

Start-LsaTlsKeyLog

Installing Wireshark

Using the installer

https://www.wireshark.org/download.html

Using chocolatey

choco install wireshark

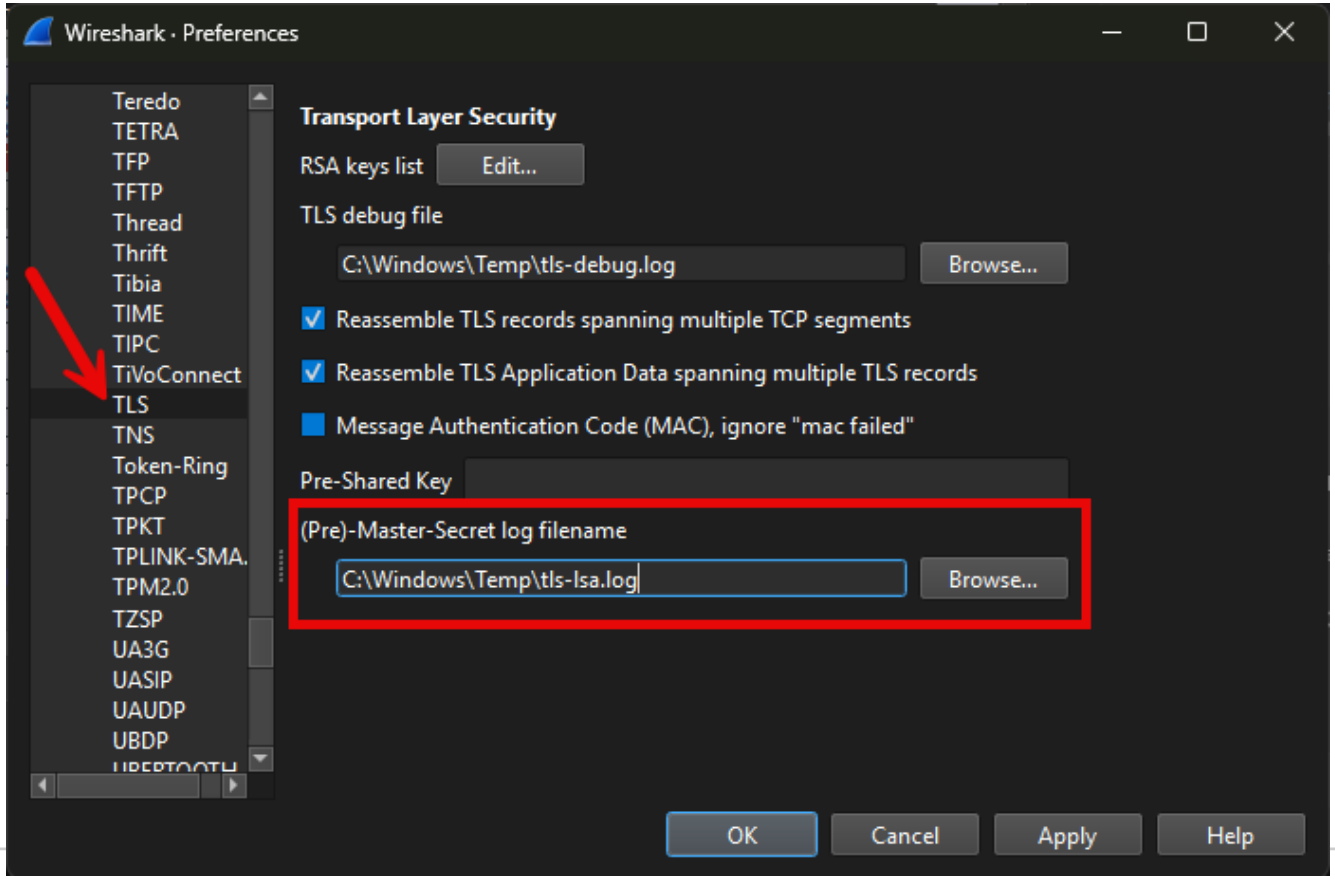
Using winget

winget install WiresharkFoundation.Wireshark

Install Wireshark in the **test virtual machine** for now

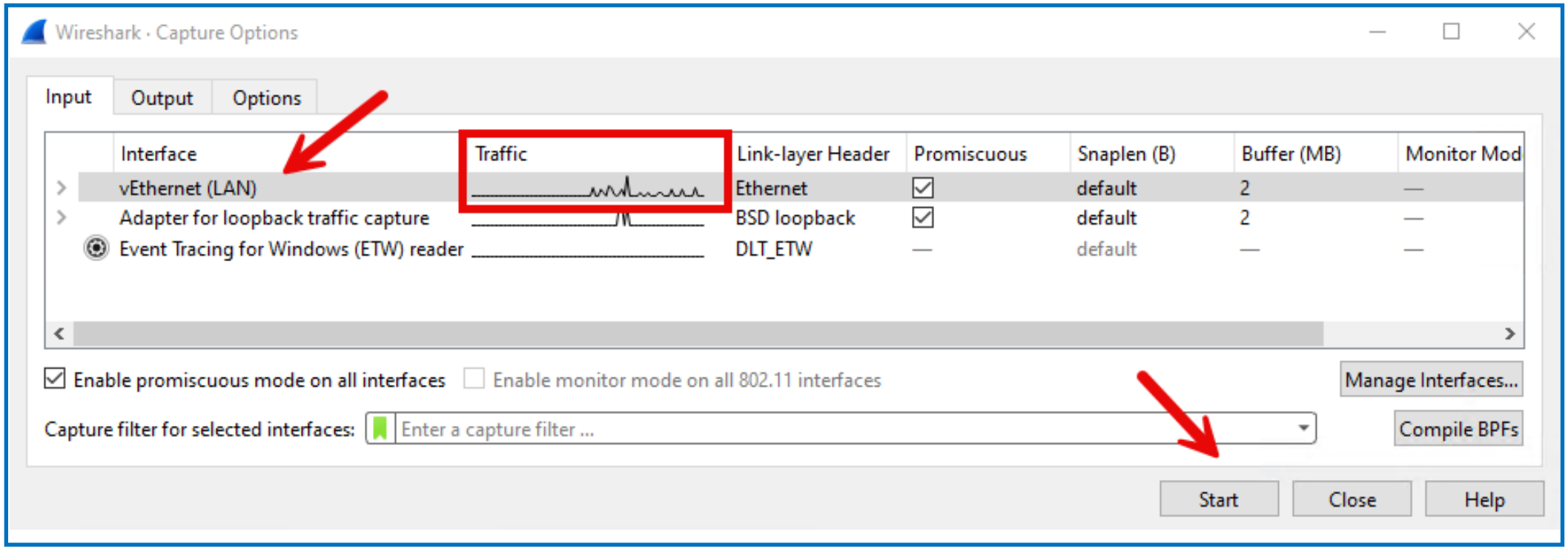
Configure Wireshark TLS Preferences

In Wireshark, open the **Preferences** dialog (Edit -> Preferences), navigate to the **TLS** section under **Protocols**, and then set the **(Pre)-Master-Secret log filename** to **“C:\Windows\Temp\tls-lsa.log”**



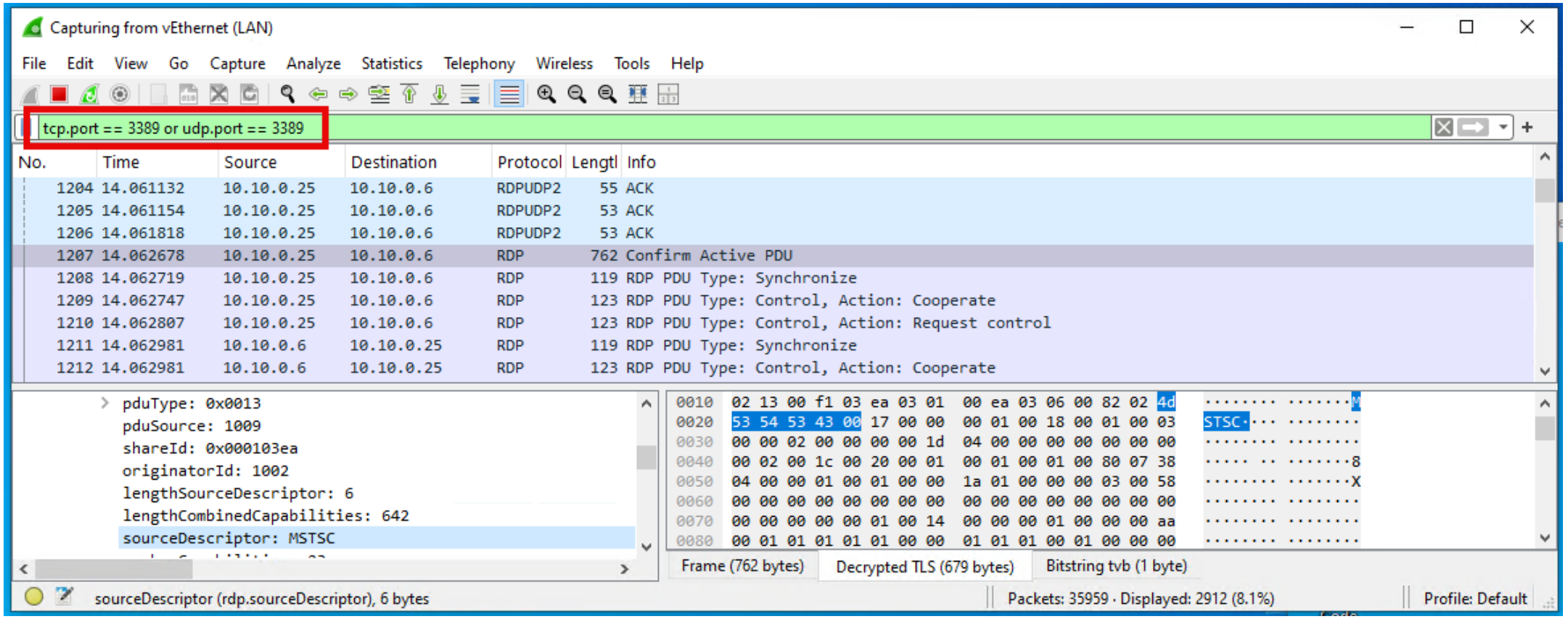
Start Wireshark Packet Capture

In Wireshark, open the **Capture Options** dialog (Capture -> Options), select the correct network interface for RDP traffic (usually the one with active traffic that isn't the loopback adapter) and then click **Start**



Wireshark Live Decrypted RDP traffic

Use “**tcp.port == 3389 or udp.port == 3389**” as display filter, then launch mstsc and connect with RDP to the VM, you should see live decrypted RDP traffic in Wireshark:



Related GitHub Repositories

Wireshark RDP resources

https://github.com/awakecoding/wireshark-rdp

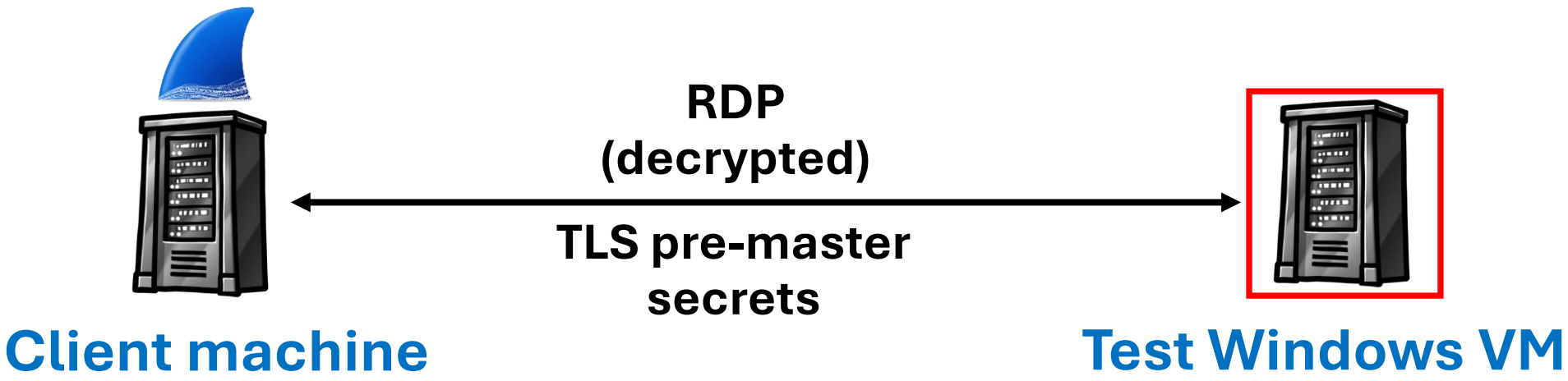
AwakeCoding Debug Tools

https://github.com/awakecoding/AwakeCoding.DebugTools

Devolutions

GOING FURTHER
with RDP traffic decryption in Wireshark

Going Further - Prerequisites



- Install Wireshark on your client machine this time
- We're going to log TLS pre-master secrets **remotely**

Logging TLS secrets remotely

In your test VM, start **TLS key log server**:

```
Start-TlsKeyLogServer -LogFile 'C:\Windows\Temp\tls-lsa.log' -Port 12345
-AllowInFirewall
```

On your host (or main device), start **TLS key log client**:

```
Install-Module AwakeCoding.DebugTools
Start-TlsKeyLogClient '10.10.0.25:12345' "$Env:Temp\tls-all.log"
```

In Wireshark, edit the TLS pre-master secrets file

You will now have TLS pre-master secrets extracted from the test VM streamed to your host, readily available for consumption in Wireshark

Logging TLS secrets from multiple sources

In each test VM (logging source):

```
Start-LsaTlsKeyLog
Start-TlsKeyLogServer -AllowInFirewall
```

On your host (or main device):

```
Start-TlsKeyLogClient @('10.10.0.10', '10.10.0.25')
"$Env:Temp\tls-all.log"
```

Logging TLS secrets when not using SChannel

With **FreeRDP**:

Use the “**/tls:secrets-file**” command-line argument:

/tls:secrets-file:C:\Users\Public\tls-freerdp.log

With **IronRDP**:

Use the “**SSLKEYLOGFILE**” environment variable:

SSLKEYLOGFILE="C:\Users\Public\tls-ironrdp.log"

Cleaning up RDP traffic for Wireshark

Disable RDP UDP transport

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services\Client' -Name 'fClientDisableUDP' -Value 1

DisableUDPTransport:i:0 (MsRdpEx only)

Disable bandwith auto-detection

connection type:i:6

networkautodetect:i:0

bandwidthautodetect:i:0

Disable bulk data compression

compression:i:0

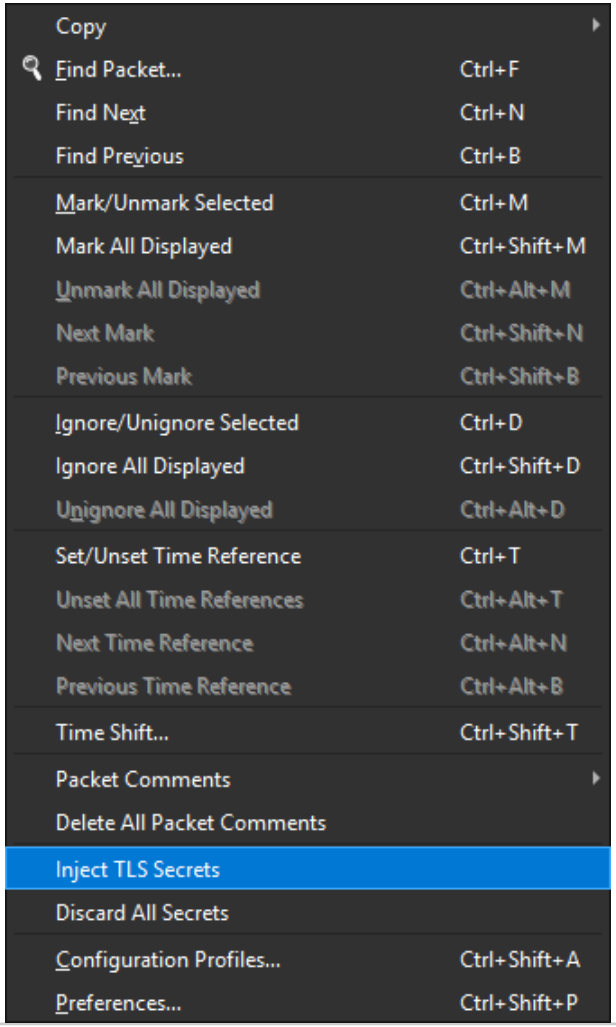
Injecting TLS secrets in capture file

In Wireshark: Edit -> **Inject TLS Secrets**

- This will embed TLS secrets inside the .pcapng file, so you no longer need the TLS key log file
- This is the best way to share decrypted packet captures – don't send TLS key log files!
- You need to do this every time before exporting a capture file, it is not automatic

Alternatively, at the command-line:

editcap.exe --inject-secrets "tls,tls-key.log" "input.pcapng" "output.pcapng"

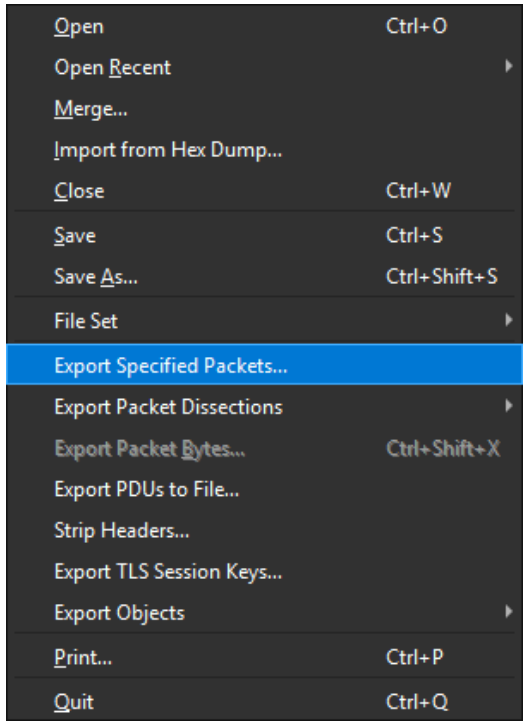


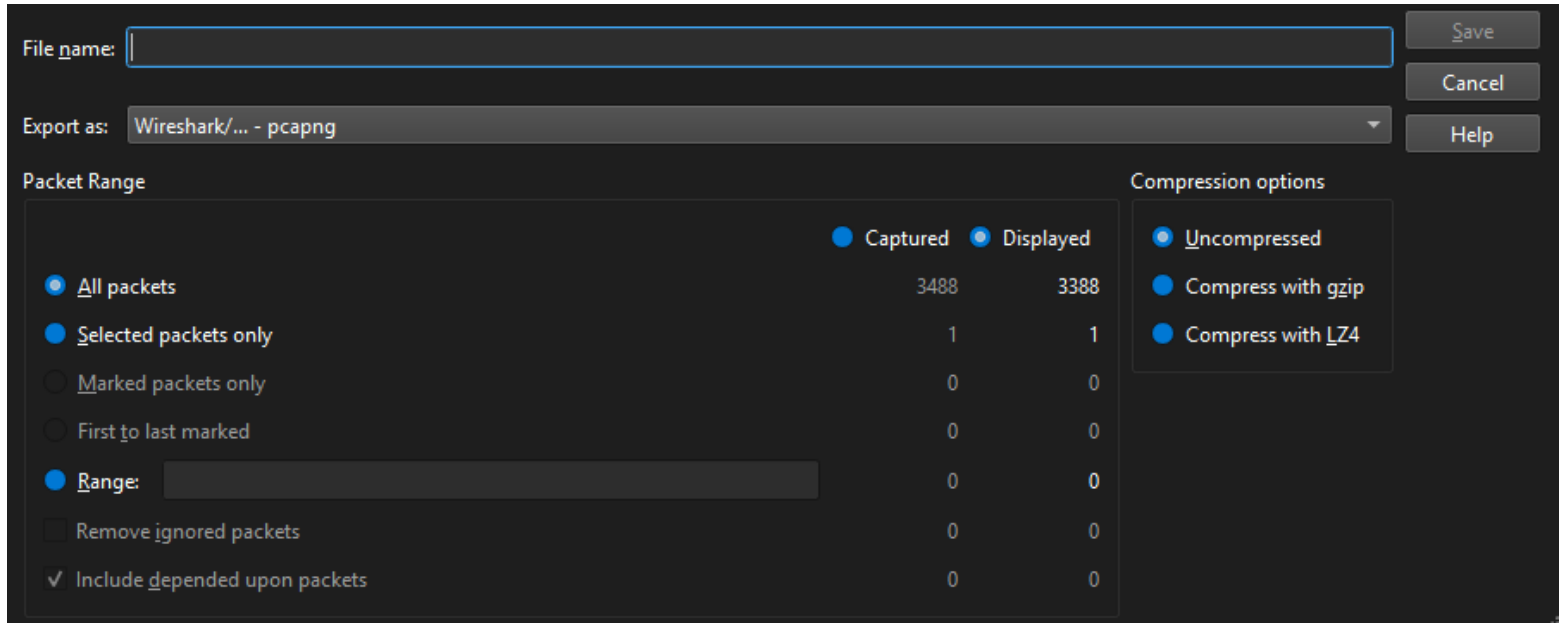
Exporting capture file

In Wireshark, File -> **Export Specified Packets…**

Select “Displayed” to export with current packet filter

Optionally select “Compress with gzip” to reduce file size





Windows packet capturing methods

npcap

Default with Wireshark - supports live capturing, but silent installation (non-GUI) requires a special license

winpcap

Deprecated and unmaintained, but still mostly works

pktmon

built-in Windows command-line tool, it would be perfect if only it could do live capturing from Wireshark

pktmon command-line capturing

Add capture filter

```
pktmon filter add RdpFilter -p 3389
```

Start capturing

```
pktmon start --capture --pkt-size 0 -f capture.etl
```

Stop capturing

```
pktmon stop
```

Export .pcapng

```
pktmon etl2pcap .\capture.etl -o capture.pcapng
```

Remove capture filter

```
pktmon filter remove RdpFilter
```

Devolutions

DEMO

Sample Wireshark RDP capture files

https://github.com/awakecoding/wireshark-rdp

wireshark-rdp/captures at master

https://github.com/awakecoding/wireshark-rdp/tree/master/captures

awakecoding / wireshark-rdp

Type / to search

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Files

master

Go to file t

captures

- README.md
- rdp-credential-guard-accepted...
- rdp-credential-guard-rejected1....
- rdp-nla-kerberos-auth1.pcapng
- rdp-nla-kerberos-auth2.pcapng
- rdp-nla-ntlm-rejected1.pcapng
- rdp-nla-ntlm-rejected2.pcapng

wireshark-rdp / captures /

Add file

Marc-André Moreau add multiple RD Gateway sample packet captures 8610ba0 · 9 months ago History

Name	Last commit message	Last commit date
..		
README.md	add multiple RD Gateway sample packet captures	9 months ago
rdp-credential-guard-accepted1.pcapng	add Restricted Admin Mode, Remote Credential ...	9 months ago
rdp-credential-guard-rejected1.pcapng	add Restricted Admin Mode, Remote Credential ...	9 months ago
rdp-nla-kerberos-auth1.pcapng	add set of sample capture files	10 months ago
rdp-nla-kerberos-auth2.pcapng	add set of sample capture files	10 months ago

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

credssp || kerberos

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
7	3.107265	10.10.0.131	10.10.0.3	KRB5	299		AS-REQ
8	3.108031	10.10.0.3	10.10.0.131	KRB5	257		KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
9	3.111392	10.10.0.131	10.10.0.3	KRB5	379		AS-REQ
10	3.111933	10.10.0.3	10.10.0.131	KRB5	1822		AS-REP
11	3.112361	10.10.0.131	10.10.0.3	KRB5	1858		TGS-REQ
12	3.113010	10.10.0.3	10.10.0.131	KRB5	1877		TGS-REP
18	3.205061	10.10.0.131	10.10.0.10	CredSSP	258		
19	3.205535	10.10.0.10	10.10.0.131	CredSSP	1434		
20	3.207185	10.10.0.131	10.10.0.3	KRB5	3120		TGS-REQ
21	3.208112	10.10.0.3	10.10.0.131	KRB5	1872		TGS-REP
22	3.208417	10.10.0.131	10.10.0.10	CredSSP	1970		

```
eric Security Service Application Program Interface
.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
rotected Negotiation
kenInit
:hTypes: 4 items
:hToken: 604f060a2a864886f7120102020304003 03fa003020105a103020110a
)5_blob: 604f060a2a864886f7120102020304003 03fa003020105a103020110a
KRB5 OID: 1.2.840.113554.1.2.2.3 (KRB5 - Kerberos 5 - User to Use
krb5_tok_id: KERB_TGT_REQUEST (0x0004)
pvno: 5
msg-type: krb-tgt-req (16)
server-name
  name-type: kRB5-NT-SRV-INST (2)
  name-string: 2 items
    KerberosString: TERMSRV
    KerberosString: IT-HELP-TEST.ad.it-help.ninja
```

```
0000  30 81 ac a0 03 02 01 06  a1 81 a4 30 81 a1 30 81   0······ ···0··0·
0010  9e a0 81 9b 04 81 98 60  81 95 06 06 2b 06 01 05   ·····` ····+···
0020  05 02 a0 81 8a 30 81 87  a0 30 30 2e 06 09 2a 86   ·····0·· ·00.··*·
0030  48 82 f7 12 01 02 02 06  09 2a 86 48 86 f7 12 01   H····· ·*·H····
0040  02 02 06 0a 2b 06 01 04  01 82 37 02 02 1e 06 0a   ····+··· ··7·····
0050  2b 06 01 04 01 82 37 02  02 0a a2 53 04 51 60 4f   +·····7· ···S·Q`O
0060  06 0a 2a 86 48 86 f7 12  01 02 02 03 04 00 30 3f   ··*·H··· ·····0?
0070  a0 03 02 01 05 a1 03 02  01 10 a2 33 30 31 a0 03   ····· ···301··
0080  02 01 02 a1 2a 30 28 1b  07 54 45 52 4d 53 52 56   ····*0(· ·TERMSRV
0090  1b 1d 49 54 2d 48 45 4c  50 2d 54 45 53 54 2e 61   ··IT-HEL P-TEST.a
00a0  64 2e 69 74 2d 68 65 6c  70 2e 6e 69 6e 6a 61      d.it-hel p.ninja
```

Frame (258 bytes) | Decrypted TLS (175 bytes)

KerberosString (kerberos.KerberosString), 29 bytes | Packets: 1178 · Displayed: 15 (1.3%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp.clientData

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
29	3.210249	10.10.0.131	10.10.0.10	RDP	545		ClientData

```
clientCoreData
    headerType: clientCoreData (0xc001)
    headerLength: 234
    versionMajor: 16
    versionMinor: 8
    desktopWidth: 1024
    desktopHeight: 768
    colorDepth: 8 bits-per-pixel (bpp) (0xca01)
    SASSequence: 43523
    keyboardLayout: 1033
    clientBuild: 22621
    clientName: IT-HELP-CLIENT
    keyboardType: IBM enhanced (101-key or 102-key) keybo
    keyboardSubType: 0
    keyboardFunctionKey: 12
    imeFileName: 00000000000000000000000000000000000000
    postBeta2ColorDepth: 8 bits-per-pixel (bpp) (0xca01)
    clientProductId: 1
    serialNumber: 0
    highColorDepth: 24-bit RGB mask (0x0018)
    supportedColorDepths: 0x000f
    earlyCapabilityFlags: 0x0f2f
    clientDigProductId: 83784cc8-01a6-4d95-98c8-38f8fae
    connectionType: LAN (10 Mbps or higher) (6)
    pad1octet: 0x00
    serverSelectedProtocol: 8
```

```
0050  ff 02 02 fc 17 02 02 ff  ff 02 01 01 02 01 00 02  ........ ........
0060  01 01 02 02 ff ff 02 01  02 04 82 01 61 00 05 00  ........ ....a...
0070  14 7c 00 01 81 58 00 08  00 10 00 01 c0 00 44 75  .|...X.. ......Du
0080  63 61 81 4a 01 c0 ea 00  10 00 08 00 00 04 00 03  ca.J.... ........
0090  01 ca 03 aa 09 04 00 00  5d 58 00 00 49 00 54 00  ........ ]X..I.T.
00a0  2d 00 48 00 45 00 4c 00  50 00 2d 00 43 00 4c 00  -.H.E.L. P.-.C.L.
00b0  49 00 45 00 4e 00 54 00  00 00 00 00 04 00 00 00  I.E.N.T. ........
00c0  00 00 00 00 0c 00 00 00  00 00 00 00 00 00 00 00  ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0100  00 00 00 00 00 00 00 00  01 ca 01 00 00 00 00 00  ........ ........
0110  18 00 0f 00 2f 0f 38 00  33 00 37 00 38 00 34 00  ..../.8. 3.7.8.4.
0120  63 00 63 00 38 00 2d 00  30 00 31 00 61 00 36 00  c.c.8.-. 0.1.a.6.
0130  2d 00 34 00 64 00 39 00  35 00 2d 00 39 00 38 00  -.4.d.9. 5.-.9.8.
0140  63 00 38 00 2d 00 33 00  38 00 66 00 38 00 66 00  c.8.-.3. 8.f.8.f.
0150  61 00 65 00 00 00 06 00  08 00 00 00 00 00 00 00  a.e..... ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 04 c0  ........ ........
0170  0c 00 15 00 00 00 00 00  00 00 02 c0 0c 00 1b 00  ........ ........
0180  00 00 00 00 00 00 03 c0  38 00 04 00 00 00 72 64  ........ 8.....rd
0190  70 64 72 00 00 00 00 00  80 80 72 64 70 73 6e 64  pdr..... ..rdpsnd
01a0  00 00 00 00 00 c0 63 6c  69 70 72 64 72 00 00 00  ......cl iprdr...
01b0  a0 c0 64 72 64 79 6e 76  63 00 00 00 80 c0 06 c0  ..drdynv c.......
01c0  08 00 00 00 00 00 0a c0  08 00 00 00 00 00         ........ ......
```

Frame (545 bytes) | Decrypted TLS (462 bytes)

clientName (rdp.client.name), 32 bytes | Packets: 1178 · Displayed: 1 (0.1%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp.channelDefArray

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
29	3.210249	10.10.0.131	10.10.0.10	RDP	545		ClientData

```
Remote Desktop Protocol
  ClientData
    clientCoreData
    clientClusterData
    clientSecurityData
    clientNetworkData
      headerType: clientNetworkData (0xc003)
      headerLength: 56
      channelCount: 4
      channelDefArray
        channelDef
          name: rdpdr
          options: 0x80800000
        channelDef
          name: rdpsnd
          options: 0xc0000000
        channelDef
          name: cliprdr
          options: 0xc0a00000
        channelDef
          name: drdynvc
          options: 0xc0800000
    clientMsgChannelData
    clientMultiTransportData
      headerType: clientMultiTransportData (0xc00a)
      headerLength: 8
      multiTransportFlags: 0x00000000
```

```
0040  00 02 01 01 02 02 04 20  02 01 02 30 1c 02 02 ff   ······ ···0····
0050  ff 02 02 fc 17 02 02 ff  ff 02 01 01 02 01 00 02   ····· ·····
0060  01 01 02 02 ff ff 02 01  02 04 82 01 61 00 05 00   ····· ····a···
0070  14 7c 00 01 81 58 00 08  00 10 00 01 c0 00 44 75   ·|···X·· ····Du
0080  63 61 81 4a 01 c0 ea 00  10 00 08 00 00 04 00 03   ca·J···· ·····
0090  01 ca 03 aa 09 04 00 00  5d 58 00 00 49 00 54 00   ····· ]X··I·T·
00a0  2d 00 48 00 45 00 4c 00  50 00 2d 00 43 00 4c 00   -·H·E·L· P·-·C·L·
00b0  49 00 45 00 4e 00 54 00  00 00 00 00 04 00 00 00   I·E·N·T· ·····
00c0  00 00 00 00 0c 00 00 00  00 00 00 00 00 00 00 00   ····· ·····
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····· ·····
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····· ·····
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····· ·····
0100  00 00 00 00 00 00 00 00  01 ca 01 00 00 00 00 00   ····· ·····
0110  18 00 0f 00 2f 0f 38 00  33 00 37 00 38 00 34 00   ····/·8· 3·7·8·4·
0120  63 00 63 00 38 00 2d 00  30 00 31 00 61 00 36 00   c·c·8·-· 0·1·a·6·
0130  2d 00 34 00 64 00 39 00  35 00 2d 00 39 00 38 00   -·4·d·9· 5·-·9·8·
0140  63 00 38 00 2d 00 33 00  38 00 66 00 38 00 66 00   c·8·-·3· 8·f·8·f·
0150  61 00 65 00 00 00 06 00  08 00 00 00 00 00 00 00   a·e····· ·····
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 04 c0   ····· ·····
0170  0c 00 15 00 00 00 00 00  00 00 02 c0 0c 00 1b 00   ····· ·····
0180  00 00 00 00 00 00 03 c0  38 00 04 00 00 00 72 64   ····· 8·····rd
0190  70 64 72 00 00 00 00 00  80 80 72 64 70 73 6e 64   pdr····· ··rdpsnd
01a0  00 00 00 00 00 c0 63 6c  69 70 72 64 72 00 00 00   ·····cl iprdr···
01b0  a0 c0 64 72 64 79 6e 76  63 00 00 00 80 c0 06 c0   ··drdynv c·····
01c0  08 00 00 00 00 00 0a c0  08 00 00 00 00 00         ····· ·····
```

Frame (545 bytes) | Decrypted TLS (462 bytes)

channelDefArray (rdp.channelDefArray), 48 bytes | Packets: 1178 · Displayed: 1 (0.1%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

t124.channelId == 1006

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
45	3.212205	10.10.0.131	10.10.0.10	T.125	95		channelJoinRequest 1006
46	3.212417	10.10.0.10	10.10.0.131	T.125	98		channelJoinConfirm 1006
104	3.337436	10.10.0.10	10.10.0.131	CLIPRDR	129		Capabilities
105	3.337436	10.10.0.10	10.10.0.131	CLIPRDR	113		Monitor ready

```
▸ Frame 46: 98 bytes on wire (784 bits), 98 bytes captured (784 bi
▸ Ethernet II, Src: Microsoft_07:7d:0a (00:15:5d:07:7d:0a), Dst: M
▸ Internet Protocol Version 4, Src: 10.10.0.10, Dst: 10.10.0.131
▸ Transmission Control Protocol, Src Port: 3389, Dst Port: 50290,
▸ Transport Layer Security
▸ TPKT, Version: 3, Length: 15
▸ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
▾ MULTIPOINT-COMMUNICATION-SERVICE T.125
  ▾ DomainMCSPDU: channelJoinConfirm (15)
    ▾ channelJoinConfirm
        result: rt-successful (0)
        initiator: 1009
        requested: 1006
        channelId: 1006
```

```
0000  03 00 00 0f 02 f0 80 3e  00 00 08 03 ee 03 ee      ·····>  ·······
```

Decrypted TLS (15 bytes)

channelId (t124.channelId), 2 bytes | Packets: 1178 · Displayed: 4 (0.3%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp.clientInfoPDU

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
49	3.213656	10.10.0.131	10.10.0.10	RDP	516		ClientInfo

```
codePage: 67699721
optionFlags: 0x0003413b
cbDomain: 0
cbUserName: 60
cbPassword: 0
cbAlternateShell: 0
cbWorkingDir: 0
domain:
userName: Administrator@ad.it-help.ninja
password:
alternateShell:
workingDir:
clientAddressFamily: 0x0002
cbClientAddress: 24
clientAddress: 10.10.0.131
cbClientDir: 64
clientDir: C:\Windows\system32\mstscax.dll
clientTimeZone
    Bias: 300
    StandardName: Eastern Standard Time
    StandardDate
    StandardBias: 0
    DaylightName: Eastern Daylight Time
    DaylightDate
    DaylightBias: 4294967236
clientSessionId: 0x00000001
```

```
0040  00 40 00 61 00 64 00 2e  00 69 00 74 00 2d 00 68   ·@·a·d·. ·i·t·-·h
0050  00 65 00 6c 00 70 00 2e  00 6e 00 69 00 6e 00 6a   ·e·l·p·. ·n·i·n·j
0060  00 61 00 00 00 00 00 00  00 00 00 02 00 18 00 31   ·a····· ·******1
0070  00 30 00 2e 00 31 00 30  00 2e 00 30 00 2e 00 31   ·0·.·1·0 ·.·0·.·1
0080  00 33 00 31 00 00 00 40  00 43 00 3a 00 5c 00 57   ·3·1···@ ·C·:·\·W
0090  00 69 00 6e 00 64 00 6f  00 77 00 73 00 5c 00 73   ·i·n·d·o ·w·s·\·s
00a0  00 79 00 73 00 74 00 65  00 6d 00 33 00 32 00 5c   ·y·s·t·e ·m·3·2·\
00b0  00 6d 00 73 00 74 00 73  00 63 00 61 00 78 00 2e   ·m·s·t·s ·c·a·x·.
00c0  00 64 00 6c 00 6c 00 00  00 2c 01 00 00 45 00 61   ·d·l·l·· ·,···E·a
00d0  00 73 00 74 00 65 00 72  00 6e 00 20 00 53 00 74   ·s·t·e·r ·n· ·S·t
00e0  00 61 00 6e 00 64 00 61  00 72 00 64 00 20 00 54   ·a·n·d·a ·r·d· ·T
00f0  00 69 00 6d 00 65 00 00  00 00 00 00 00 00 00 00   ·i·m·e·· ********
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 0b   ******** ********
0110  00 00 00 01 00 02 00 00  00 00 00 00 00 00 00 00   ******** ********
0120  00 45 00 61 00 73 00 74  00 65 00 72 00 6e 00 20   ·E·a·s·t ·e·r·n·
0130  00 44 00 61 00 79 00 6c  00 69 00 67 00 68 00 74   ·D·a·y·l ·i·g·h·t
0140  00 20 00 54 00 69 00 6d  00 65 00 00 00 00 00 00   · ·T·i·m ·e******
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ******** ********
0160  00 00 00 03 00 00 00 02  00 02 00 00 00 00 00 00   ******** ********
0170  00 c4 ff ff ff 01 00 00  00 86 01 00 00 00 00 64   ******** *******d
0180  00 00 00 2a 00 45 00 61  00 73 00 74 00 65 00 72   ***·*·E·a ·s·t·e·r
0190  00 6e 00 20 00 53 00 74  00 61 00 6e 00 64 00 61   ·n· ·S·t ·a·n·d·a
01a0  00 72 00 64 00 20 00 54  00 69 00 6d 00 65 00 00   ·r·d· ·T ·i·m·e··
01b0  00                                                  ·
```

Frame (516 bytes) | Decrypted TLS (433 bytes) | Bitstring tvb (1 byte)

StandardName (rdp.Name.Standard), 64 bytes | Packets: 1178 · Displayed: 1 (0.1%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp.capabilitySet

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
51	3.236325	10.10.0.10	10.10.0.131	RDP	555		Demand Active PDU
53	3.237947	10.10.0.131	10.10.0.10	RDP	694		Confirm Active PDU

```
sourceDescriptor: MSTSC
numberCapabilities: 23
pad2Octets: 0
General
Bitmap
Order
Bitmap Cache Rev 2
Color Cache
Activation
Control
Pointer
Share
Input
Sound
Font
Glyph Cache
Brush
Off-screen Cache
Virtual Channel
Draw Nine Grid Cache
Multi-Fragment Update
Surface Commands
Large Pointer
Frame acknowledge
Window
Bitmap Codecs
```

```
0010  02 13 00 f1 03 ea 03 01  00 ea 03 06 00 3e 02 4d  ........ .....>.M
0020  53 54 53 43 00 17 00 00  00 01 00 18 00 01 00 03  STSC.... ........
0030  00 00 02 00 00 00 00 1d  04 00 00 00 00 00 00 00  ........ ........
0040  00 02 00 1c 00 20 00 01  00 01 00 01 00 00 04 00  ..... .. ........
0050  03 00 00 01 00 01 00 00  1a 01 00 00 00 03 00 58  ........ .......X
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0070  00 00 00 00 00 01 00 14  00 00 00 01 00 00 00 aa  ........ ........
0080  00 01 01 01 01 01 00 00  01 01 01 00 01 00 00 00  ........ ........
0090  01 01 01 01 01 01 01 01  00 01 01 01 00 00 00 00  ........ ........
00a0  00 a1 06 06 00 00 00 00  00 00 84 03 00 00 00 00  ........ ........
00b0  00 e4 04 00 00 13 00 28  00 02 00 00 03 78 00 00  .......( .....x..
00c0  00 78 00 00 00 51 01 00  00 00 00 00 00 00 00 00  .x...Q.. ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 0a 00 08  ........ ........
00e0  00 06 00 00 00 07 00 0c  00 00 00 00 00 00 00 00  ........ ........
00f0  00 05 00 0c 00 00 00 00  00 02 00 02 00 08 00 0a  ........ ........
0100  00 01 00 14 00 15 00 09  00 08 00 00 00 00 00 0d  ........ ........
0110  00 58 00 b1 00 00 00 09  04 00 00 04 00 00 00 00  .X...... ........
0120  00 00 00 0c 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0160  00 00 00 00 00 00 00 0c  00 08 00 01 00 00 00 0e  ........ ........
0170  00 08 00 01 00 00 00 10  00 34 00 fe 00 04 00 fe  ........ .4......
0180  00 04 00 fe 00 08 00 fe  00 08 00 fe 00 10 00 fe  ........ ........
```

Frame (694 bytes) | Decrypted TLS (611 bytes) | Bitstring tvb (1 byte)

sourceDescriptor (rdp.sourceDescriptor), 6 bytes | Packets: 1178 · Displayed: 2 (0.2%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp_drdynvc.cmd == 1

No.	Time	Source	Destination	Protocol	Length	PDU type	Channel Name	Info
67	3.239238	10.10.0.10	10.10.0.131	DRDYNVC	142	Create PDU	Microsoft::Windows::RDS::Telemetry	CreateChan
68	3.239380	10.10.0.10	10.10.0.131	DRDYNVC	141	Create PDU	Microsoft::Windows::RDS::Graphics	CreateChan
72	3.239623	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
74	3.240926	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
79	3.264028	10.10.0.10	10.10.0.131	DRDYNVC	155	Create PDU	Microsoft::Windows::RDS::Video::Control::v08.01	CreateChan
80	3.264028	10.10.0.10	10.10.0.131	DRDYNVC	152	Create PDU	Microsoft::Windows::RDS::Video::Data::v08.01	CreateChan
81	3.264028	10.10.0.10	10.10.0.131	DRDYNVC	149	Create PDU	Microsoft::Windows::RDS::Geometry::v08.01	CreateChan
83	3.264631	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
84	3.265385	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
88	3.265547	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
100	3.328156	10.10.0.10	10.10.0.131	DRDYNVC	149	Create PDU	Microsoft::Windows::RDS::Geometry::v08.01	CreateChan
101	3.328695	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
102	3.330446	10.10.0.10	10.10.0.131	DRDYNVC	138	Create PDU	Microsoft::Windows::RDS::Input	CreateChan
103	3.330841	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
113	3.387289	10.10.0.10	10.10.0.131	DRDYNVC	149	Create PDU	Microsoft::Windows::RDS::Geometry::v08.01	CreateChan
114	3.387564	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan
116	3.439089	10.10.0.10	10.10.0.131	DRDYNVC	147	Create PDU	Microsoft::Windows::RDS::DisplayControl	CreateChan
117	3.439724	10.10.0.131	10.10.0.10	DRDYNVC	111	Create PDU		CreateChan

```
▸ MULTIPOINT-COMMUNICATION-SERVICE T.125
▾ Remote Desktop Protocol
  ▾ SendData
    ▸ channelPDUHeader: 2500000003000000
▾ RDP Dynamic Channel Protocol
    .... ..00 = ChannelId length: 1 byte (0x0)
    .... 10.. = Pri: PriorityCharge2 (0x2)
    0001 .... = PDU type: Create PDU (0x1)
    Channel Id: 0x00000005
    Channel Name: Microsoft::Windows::RDS::Telemetry
```

```
0000  03 00 00 3b 02 f0 80 68  00 01 03 ef f0 2d 25 00   ···;···h ·····-%·
0010  00 00 03 00 00 00 18 05  4d 69 63 72 6f 73 6f 66   ····· Microsof
0020  74 3a 3a 57 69 6e 64 6f  77 73 3a 3a 52 44 53 3a   t::Windo ws::RDS:
0030  3a 54 65 6c 65 6d 65 74  72 79 00                  :Telemet ry·
```

Frame (142 bytes) | Decrypted TLS (59 bytes) | Bitstring tvb (1 byte)

Channel Name (rdp_drdynvc.channelName), 35 bytes | Packets: 1178 · Displayed: 24 (2.0%) | Profile: Default

rdp-nla-kerberos-auth1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rdp_egfx

No.	Time	Source	Destination	Protocol	Length	PDU type	Info
77	3.262302	10.10.0.131	10.10.0.10	EGFX	226	Data PDU	Caps advertise
78	3.263891	10.10.0.10	10.10.0.131	EGFX	123	Data PDU	Caps confirm
85	3.265396	10.10.0.10	10.10.0.131	EGFX	150	Data PDU	Reset graphics,Start frame,End frame
98	3.275540	10.10.0.131	10.10.0.10	EGFX	127	Data PDU	Frame acknowledge
144	3.479192	10.10.0.10	10.10.0.131	EGFX	904	Data PDU	Create Surface,Map Surface To Scaled Output,Start frame,So
145	3.483417	10.10.0.131	10.10.0.10	EGFX	127	Data PDU	Frame acknowledge
171	3.517700	10.10.0.10	10.10.0.131	EGFX	9156	Data PDU,D…	Surface To Cache,Cache To Surface,Cache To Surface,Cache T
184	3.523110	10.10.0.131	10.10.0.10	EGFX	127	Data PDU	Frame acknowledge

```
    [ChannelName: Microsoft::Windows::RDS::Graphics]
    [DataProgress: 0-796/796]
▾ RDP Graphic pipeline channel Protocol
    CmdId: Create surface (0x0009)
    flags: 0x0000
    pduLength: 15
▾ RDP Graphic pipeline channel Protocol
    CmdId: Map surface to scaled output (0x0017)
    flags: 0x0000
    pduLength: 28
▾ RDP Graphic pipeline channel Protocol
    CmdId: Start frame (0x000b)
    flags: 0x0000
    pduLength: 16
    Timestamp: 0
    Frame id: 0x00000002
    [Frame acked in: 145]
▾ RDP Graphic pipeline channel Protocol
    CmdId: Solid fill (0x0004)
    flags: 0x0000
    pduLength: 24
```

```
0000  09 00 00 00 0f 00 00 00  00 00 00 04 00 03 21 17  ........ ......!.
0010  00 00 00 1c 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
0020  00 00 00 00 04 00 00 00  03 00 00 0b 00 00 00 10  ........ ........
0030  00 00 00 00 00 00 00 02  00 00 00 04 00 00 00 18  ........ ........
0040  00 00 00 00 00 00 00 00  00 01 00 00 00 00 00 40  ........ .......@
0050  00 40 00 06 00 00 00 1c  00 00 00 00 00 f6 17 b0  .@...... ........
0060  bf 6e 5f ce a9 02 00 00  00 00 00 40 00 40 00 07  .n_..... ...@.@..
0070  00 00 00 12 00 00 00 02  00 00 00 01 00 40 00 00  ........ .....@..
0080  00 07 00 00 00 12 00 00  00 02 00 00 00 01 00 80  ........ ........
0090  00 00 00 07 00 00 00 12  00 00 00 02 00 00 00 01  ........ ........
00a0  00 c0 00 00 00 07 00 00  00 12 00 00 00 02 00 00  ........ ........
00b0  00 01 00 00 01 00 00 07  00 00 00 12 00 00 00 02  ........ ........
00c0  00 00 00 01 00 40 01 00  00 07 00 00 00 12 00 00  .....@.. ........
00d0  00 02 00 00 00 01 00 80  01 00 00 07 00 00 00 12  ........ ........
00e0  00 00 00 02 00 00 00 01  00 c0 01 00 00 07 00 00  ........ ........
00f0  00 12 00 00 00 02 00 00  00 01 00 00 02 00 00 07  ........ ........
0100  00 00 00 12 00 00 00 02  00 00 00 01 00 40 02 00  ........ .....@..
0110  00 07 00 00 00 12 00 00  00 02 00 00 00 01 00 80  ........ ........
0120  02 00 00 07 00 00 00 12  00 00 00 02 00 00 00 01  ........ ........
```

res) | Decrypted TLS (821 bytes) | Bitstring tvb (1 byte) | Uncompressed GFX (3561 bytes)

RDP Graphic pipeline channel Protocol (rdp_egfx), 15 bytes | Packets: 1178 · Displayed: 90 (7.6%) | Profile: Default

Devolutions

QUESTIONS?

Devolutions

Thank you!

https://devolutions.net
https://bsky.app/profile/devolutions.net

https://awakecoding.com
https://bsky.app/profile/awakecoding.com