RDP 协议格式详解

作者：Keary

鉴于微软至今未公布 RDP 协议文档，网上资料又与实际抓包有不少出入，故根据自己分析进行总结。

首先根据以下范例表，对本文如何表示协议字段格式做简单介绍：

C2 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0013(19)

表 1 RDP 字段格式范例表

如范例表所示，左上角的 C2 代表 RDP 连接之后，客户端（C）发送的第二个数据包，若为 S3 则表示服务器（S）发送的第三个数据包；

表右上角的“TPKT from Client”表示该表的含义。

表中 Offset 表示该字段相对首部的偏移量；

表中 Datatype 表示该字段的数据类型，Int16 表示以 16bit 的整型变量表示该字段，be 表示以高位开始存储数据（大端），le 表示低位开始存储数据（小端）；在网络数据包中，单个字节内的比特顺序不受大端小端影响，故 Int8 没有提示高位还是低位存储。

表中 Description 表示对该字段含义的解释；

表中 Expect Value 表示该字段的预计值，若表的该字段有值，则表示在一般情况下，该字段即为表中的值；

表中 Value 表示我们抓包获得的实际值。

我们捕获并分析数据包的测试环境如下：客户端 Win7 SP1(RDP 7.1)；服务器 Win2003（RDP 5.2），取消了身份验证功能，加密等级为“高”。

一. RDP 连接阶段

1. ISO 层连接

在客户端与服务器建立完 TCP 连接后，首先需要建立 RDP 的 ISO 层连接。要注意的是，在整个连接阶段，RDP 协议采用 TPKT 协议进行封装。

客户端首先发送 ISO 层连接请求，服务器收到以后，向客户端发送 ISO 连接确认，至此 ISO 层连接建立完毕。

这两个数据包如下表所示：

C1 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0013(19)
4	Variable Length	TPDU		见 C1-0

表 2

C1-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x0e(14)
1	Int8	TPDU Packet Type	0xe0	0xe0(224)
2	Int16(be)	Dst Ref		0x0000
4	Int16(be)	Src Ref		0x0000
6	Int8	Class		0x00
7	Variable Length(le)	RDP Routing Token		0x0000000000080001

表 3

“TPKT Version”为 TPKT 协议的版本号，该值永远为 3；“TPKT Reserved”为保留字段，永远为 0；“TPKT Length”为 TPKT 协议长度，该值实际上等于整个应用层数据的长度；

TPDU 为 TPKT 协议的数据部分，其字段具体含义：“TPDU Length”值实际上等于从下一个字段开始算的长度（不包括子包长度）；“TPDU Packet Type”表示 TPDU 代表的类型，0xe0 连接请求，0xd0 连接确认，0xf0 数据，0x80 断开连接请求，0x70 错误；“Dst Ref”和“Src Ref”含义未知，一般都为 0；“Class”

含义未知，一般为 0；“RDP Routing Token”同样含义不明确，可能与 Active X 控件有关，有时其值为类似“Cookie: mstshash=A70067”的文字。

服务器收到 ISO 连接请求后，发送连接确认数据包：

S1 TPKT from Server

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0013(19)
4	Variable Length	TPDU		见 S1-0

表 4

S1-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x0e(14)
1	Int8	TPDU Packet Type	0xd0	0xd0(208)
2	Int16(be)	Dst Ref		0x0000
4	Int16(be)	Src Ref		0x1234(4660)
6	Int8	Class		0x00
7	Variable Length(le)	RDP Routing Token		0x0000000000080002

表 5

服务器发送的 ISO 确认包与 ISO 请求包格式类似，不再多做介绍。

2. MCS 层连接

ISO 连接建立完毕后，接下来建立 MCS 层连接。MCS 层连接分为两大部分，分别是“交换基本设置连接”，和“通道连接”。

MCS 层连接阶段的头两个数据包即为“交换基本设置连接”，即客户端发送的“MCS Connect Initial Packet”和服务器发送的“MSC Connect Response Packet”。

“通道连接”由多个数据包组成，主要功能是协商域、用户和通道设置。

以下两个数据包为“交换基本设置连接”的字段格式：

C2 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x01ac(428)
4	Variable Length	TPDU		见 C2-0

表 6

C2-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Initial Packet		见 C2-0-0

表 7

C2-0-0 MCS Initial Packet

Offset	Datatype	Description	Expect Value	Value
0	Int16(be)	MCS Initial Tag	0x7f65	0x7f65(32613)
2	Int24(be)	MCS Length	0x820000 | Length	0x8201a0
5	Int8	BER_TAG_OCTET_STRING	0x04	0x04
6	Int8	BER_TAG_LENGTH	0x01	0x01
7	Int8	Calling Domain Selector	0x01	0x01
8	Int8	BER_TAG_OCTET_STRING	0x04	0x04
9	Int8	BER_TAG_LENGTH	0x01	0x01
10	Int8	Called Domain Selector	0x01	0x01

11	Int8	BER_TAG_BOOLEAN	0x01	0x01
12	Int8	BER_TAG_LENGTH	0x01	0x01
13	Int8	Upward Flag	0xff	0xff
14	Variable Length	Target Parameters		**见 C2-0-0-0**
41	Variable Length	Min Parameters		**见 C2-0-0-1**
68	Variable Length	Max Parameters		**见 C2-0-0-2**
98	Variable Length	MCS Initial Data		**见 C2-0-0-3**

表 8

MCS 包的头一个字段为标签字段，指示该 MCS 包的类别，值为 0x7f65 表示该包为 MCS 初始化请求包，0x7f66 表示 MCS 初始化响应包；

“MCS Length”的值为从下一个字段开始所有数据的长度（包括子数据包）；

MCS 包从第三个字段开始，是由多个 BER 段落组成，每个 BER 段落的结构如下：（BER 标签，BER 长度，BER 内容），其中 BER 长度仅指本 BER 的内容长度。

BER 标签可能的取值为：1/BER_TAG_BOOLEAN，2/BER_TAG_INTEGER，4/BER_TAG_OCTET_STRING，10/BER_TAG_RESULT，0x30/MCS_TAG_DOMAIN_PARAMS。

C2-0-0-0 MCS Target Parameters

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_DOMAIN_PARAMETERS	0x30	0x30(48)
1	Int8	BER_TAG_LENGTH		0x19
2	Int8	BER_TAG_INTEGER	0x02	0x02
3	Int8	BER_TAG_LENGTH		0x01
4	Int8	Channels Value		0x22(34)
5	Int8	BER_TAG_INTEGER	0x02	0x02
6	Int8	BER_TAG_LENGTH		0x01
7	Int8	Users Value		0x02
8	Int8	BER_TAG_INTEGER	0x02	0x02
9	Int8	BER_TAG_LENGTH		0x01
10	Int8	Tokens Value		0x00

11	Int8	BER_TAG_INTEGER	0x02	0x02
12	Int8	BER_TAG_LENGTH		0x01
13	Int8	Priorities Value		0x01
14	Int8	BER_TAG_INTEGER	0x02	0x02
15	Int8	BER_TAG_LENGTH		0x01
16	Int8	Throughput Value		0x00
17	Int8	BER_TAG_INTEGER	0x02	0x02
18	Int8	BER_TAG_LENGTH		0x01
19	Int8	Height Value		0x01
20	Int8	BER_TAG_INTEGER	0x02	0x02
21	Int8	BER_TAG_LENGTH		0x02
22	Int16(be)	PDU Max Size		0xffff(65535)
24	Int8	BER_TAG_INTEGER	0x02	0x02
25	Int8	BER_TAG_LENGTH		0x01
26	Int8	Protocol Version		0x02

表 9

表 9 所示的子数据包是客户端向服务器推荐的通道、优先级等值的设置 。表 10、11 则是客户端所允许的此类值的最小和最大值。注意，第二个字段的 BER_TAG_LENGTH 指的是该子数据包内其它所有 BER 段的总长度（因为该子数据包内的其它 BER 段都属于第一个 BER 段的内容）。

C2-0-0-1 MCS Min Parameters

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_DOMAIN_PARAMETERS	0x30	0x30(48)
1	Int8	BER_TAG_LENGTH		0x19
2	Int8	BER_TAG_INTEGER	0x02	0x02
3	Int8	BER_TAG_LENGTH		0x01
4	Int8	Channels Value		0x01
5	Int8	BER_TAG_INTEGER	0x02	0x02

6	Int8	BER_TAG_LENGTH		0x01
7	Int8	Users Value		0x01
8	Int8	BER_TAG_INTEGER	0x02	0x02
9	Int8	BER_TAG_LENGTH		0x01
10	Int8	Tokens Value		0x01
11	Int8	BER_TAG_INTEGER	0x02	0x02
12	Int8	BER_TAG_LENGTH		0x01
13	Int8	Priorities Value		0x01
14	Int8	BER_TAG_INTEGER	0x02	0x02
15	Int8	BER_TAG_LENGTH		0x01
16	Int8	Throughput Value		0x00
17	Int8	BER_TAG_INTEGER	0x02	0x02
18	Int8	BER_TAG_LENGTH		0x01
19	Int8	Height Value		0x01
20	Int8	BER_TAG_INTEGER	0x02	0x02
21	Int8	BER_TAG_LENGTH		0x02
22	Int16(be)	PDU Max Size		0x0420(1056)
24	Int8	BER_TAG_INTEGER	0x02	0x02
25	Int8	BER_TAG_LENGTH		0x01
26	Int8	Protocol Version		0x02

表 10

C2-0-0-2 MCS Max Parameters

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_DOMAIN_PARAMETERS	0x30	0x30(48)
1	Int8	BER_TAG_LENGTH		0x1c
2	Int8	BER_TAG_INTEGER	0x02	0x02
3	Int8	BER_TAG_LENGTH		0x02
4	Int16(be)	Channels Value		0xffff

6	Int8	BER_TAG_INTEGER	0x02	0x02
7	Int8	BER_TAG_LENGTH		0x02
8	Int16(be)	Users Value		0xfc17
10	Int8	BER_TAG_INTEGER	0x02	0x02
11	Int8	BER_TAG_LENGTH		0x02
12	Int16(be)	Tokens Value		0xffff
14	Int8	BER_TAG_INTEGER	0x02	0x02
15	Int8	BER_TAG_LENGTH		0x01
16	Int8	Priorities Value		0x01
17	Int8	BER_TAG_INTEGER	0x02	0x02
18	Int8	BER_TAG_LENGTH		0x01
19	Int8	Throughput Value		0x00
20	Int8	BER_TAG_INTEGER	0x02	0x02
21	Int8	BER_TAG_LENGTH		0x01
22	Int8	Height Value		0x01
23	Int8	BER_TAG_INTEGER	0x02	0x02
24	Int8	BER_TAG_LENGTH		0x02
25	Int16(be)	PDU Max Size		0xffff(65535)
27	Int8	BER_TAG_INTEGER	0x02	0x02
28	Int8	BER_TAG_LENGTH		0x01
29	Int8	Protocol Version		0x02

表 11

C2-0-0-3 MCS Initial Data

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_OCTET_STRING	0x04	0x04
1	Int24(be)	MCS Initial Data Length	0x820000 | Length	0x82013f
4	Int16(be)	Unknown	0x0005	0x0005

6	Int16(be)	Unknown	0x0014	0x0014
8	Int8	Unknown	0x7c	0x7c
9	Int16(be)	Unknown	0x0001	0x0001
11	Int16(be)	Remaining Length	0x8000 | Length	0x8136
13	Int16(be)	Unknown	0x0008	0x0008
15	Int16(be)	Unknown	0x0010	0x0010
17	Int8	Unknown	0x00	0x00
18	Int16(be)	Unknown	0x01c0	0x01c0
20	Int8	Unknown	0x00	0x00
21	Int32(le)	OEM ID:"Duca"	0x61637544	0x61637544
25	Int16(be)	Remaining Length	0x8000 | Length	0x8128
27	Variable Length	Client Info		**见 C2-0-0-3-0**
247	Variable Length	Client Sec Info		**见 C2-0-0-3-1**
259	Variable Length	Client Encryption Settings		**见 C2-0-0-3-2**
271	Variable Length	Client Channels Info		**见 C2-0-0-3-3**

表 12

表 12 中“MCS Initial Data Length”和“Remaining Length”都是指从该字段的下一个字段开始，直至数据包结束的所有内容长度（包括子数据包，注意并不直接等于长度，而是参与“或运算”后的值）。

C2-0-0-3-0 Client Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Client Info Tag	0xc001	0xc001
2	Int16(le)	Client Info Length		0x00d8
4	Int16(le)	RDP Version	1 for RDP4, 4 for RDP5	0x0004
6	Int16(le)	Unknown	0x0008	0x0008
8	Int16(le)	Screen Width		0x0500

10	Int16(le)	Screen Height		0x0400
12	Int16(le)	Unknown	0xca01	0xca01
14	Int16(le)	Unknown	0xaa03	0xaa03
16	Int32(le)	Keylayout		0x00000804
20	Int32(le)	Client Build		0x00001db1
24	Unicode String(32Byte)	Host Name		"KEARY-PC"
56	Int32(le)	Keyboard Type	0x00000004	0x00000004
60	Int32(le)	Keyboard Subtype	0x00000000	0x00000000
64	Int32(le)	Keyboard Functionkeys	0x0000000c	0x0000000c
68	64Byte	Reserved Data	0	0
132	Int16(le)	Color Depth		0xca01(51713) (8bits depth)
134	Int16(le)	Unknown	0x0001	0x0001
136	Int32(le)	Unknown	0x00000000	0x00000000
140	Int8	Server Depth		0x18
141	Int16(le)	Unknown		0x0f00
143	Int24(le)	Unknown		0x002f00
146	Unicode String(64Byte)	Unknown		"dcb2232b-cc60-40f8-bbd7-16bd738"
210	Int8	Unknown		0x02
211	(Client_Info_Leng - 211)Byte	Reserved		0x0000000000

表 13

表 13 第一个字段表示该子数据包的含义（即 TAG 字段），可能的取值有：0xc001/SEC_TAG_CLI_INFO ，0xc002/SEC_TAG_CLI_CRYPT ，0xc003/SEC_TAG_CLI_CHANNELS，0xc004/SEC_TAG_CLI_4。由于该子数据包为客

户端信息类型，故取值为 0xc001。

“Client Info Length”值为整个子数据包的长度（这点跟 BER 段有点不同）。该表中最后一个字段的内容为全 0，其长度则取决于“Client Info Length”的值，即长度等于：（Client Info Length－211）Byte，在我们实际抓包中，Client Info Length 等于 0xd8(216)，故此字段长度为 5Byte。

C2-0-0-3-1 Client Sec Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Client Sec Tag	0xc004	0xc004
2	Int16(le)	Client Sec Length		0x000c
4	Int32(le)	Unknown		0x00000011
8	Int32(le)	Reserved	0	0x00000000

表 14

表 14 中“Client Sec Length”值为整个子数据包的长度。第三个字段含义不确定，其值一般只可能取 0x09、0x0b、0x11 三个值。

C2-0-0-3-2 Client Encryption Settings

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Client Encryption Tag	0xc002	0xc002
2	Int16(le)	Client Encryption Length		0x000c
4	Int32(le)	Unknown		0x0000001b
8	Int32(le)	Reserved	0	0x00000000

表 15

表 15 与表 14 类似，其中第三个字段可能是客户端向服务器建议的 RC4 加密等级（如 40bit,56bit,128bit,FIPS 等），其值一般等于 0x1b 或 0x0b，若令其等于 0 则表示不希望使用加密。

C2-0-0-3-3 Client Channels Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Client Channels Tag	0xc003	0xc003

2	Int16(le)	Client Channels Length	(Channels Num) * 12 + 8	0x0038
4	Int32(le)	Channels Num		0x00000004
8	String(8Byte)	Channel Name		"rdpdr"
16	Int32(be)	Channel Flags		0x00008080
20	String(8Byte)	Channel Name		"rdpsnd"
28	Int32(be)	Channel Flags		0x000000c0
32	String(8Byte)	Channel Name		"drdynvc"
40	Int32(be)	Channel Flags		0x000080c0
44	String(8Byte)	Channel Name		"cliprdr"
52	Int32(be)	Channel Flags		0x0000a0c0

表 16

“Client Channels Length”值为整个子数据包的长度，其计算公式为：通道数 * 12 + 8，其中通道数即第三个字段“Channels Num”的值。

从第四个字段开始，是由多个通道数据段组成，每个通道数据段的格式如下：（通道名 8Byte，通道标签 4Byte），不同的 RDP 连接环境可能通道数也不同，因此反映在数据包中就是字段的个数也不同，但每个通道数据段的格式是固定的（在本例中，申请了 4 个通道，故其后有 8 个字段）。

S2 TPKT from Server

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0151(337)
4	Variable Length	TPDU		见 S2-0

表 17

S2-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)

2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Response Packet		见 S2-0-0

表 18

S2-0-0 MCS Response Packet

Offset	Datatype	Description	Expect Value	Value
0	Int16(be)	MCS Response Tag	0x7f66	0x7f66
2	Int24(be)	MCS Length	0x820000 | Length	0x820145
5	Int8	BER_TAG_RESULT	0x0a	0x0a
6	Int8	BER_TAG_LENGTH		0x01
7	Int8	Result Value	0x00	0x00
8	Int8	BER_TAG_INTEGER	0x02	0x02
9	Int8	BER_TAG_LENGTH		0x01
10	Int8	Connect ID	0x00	0x00
11	Variable Length	Domain Parameters		见 S2-0-0-0
39	Variable Length	User Data		见 S2-0-0-1

表 19

表 19 中“MCS Length”值为从下一个字段开始直至数据包结束的长度（包括子数据包）；“Result Value”含义应该是指上一步的连接是否成功，值为 0 即成功；“Domain Parameters”是服务器根据客户端发来的参数建议值、最大值和最小值三组参数权衡后，给出的最终参数设置。

S2-0-0-0 Domain Parameters

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_DOMAIN_PARAMETERS	0x30	0x30(48)
1	Int8	BER_TAG_LENGTH		0x1a
2	Int8	BER_TAG_INTEGER	0x02	0x02
3	Int8	BER_TAG_LENGTH		0x01
4	Int8	Channels Value		0x22(34)

5	Int8	BER_TAG_INTEGER	0x02	0x02
6	Int8	BER_TAG_LENGTH		0x01
7	Int8	Users Value		0x03
8	Int8	BER_TAG_INTEGER	0x02	0x02
9	Int8	BER_TAG_LENGTH		0x01
10	Int8	Tokens Value		0x00
11	Int8	BER_TAG_INTEGER	0x02	0x02
12	Int8	BER_TAG_LENGTH		0x01
13	Int8	Priorities Value		0x01
14	Int8	BER_TAG_INTEGER	0x02	0x02
15	Int8	BER_TAG_LENGTH		0x01
16	Int8	Throughput Value		0x00
17	Int8	BER_TAG_INTEGER	0x02	0x02
18	Int8	BER_TAG_LENGTH		0x01
19	Int8	Height Value		0x01
20	Int8	BER_TAG_INTEGER	0x02	0x02
21	Int8	BER_TAG_LENGTH		0x03
22	Int24(be)	PDU Max Size		0x00fff8(65528)
25	Int8	BER_TAG_INTEGER	0x02	0x02
26	Int8	BER_TAG_LENGTH		0x01
27	Int8	Protocol Version		0x02

表 20

S2-0-0-1 User Data

Offset	Datatype	Description	Expect Value	Value
0	Int8	BER_TAG_OCTET_STRING	0x04	0x04
1	Int24(be)	BER_TAG_LENGTH	0x820000 | Length	0x82011f
4	21Byte	Unknown		0x00500014...
25	Int16(be)	Unknown		0x8108

27	Variable Length	Server Info		见 S2-0-0-1-0
39	Variable Length	Server Channels Info		见 S2-0-0-1-1
55	Variable Length	Server Encrypt Info		见 S2-0-0-1-2

表 21

表 21 中“BER_TAG_LENGTH”值为下一个字段开始直至数据包结束的长度（包括子数据包）；第三个字段的“Unknown”为固定 21 字节的信息，含义可能是 T.124 协议的头部；第四个字段的“Unknown”在一些资料上说其为长度信息，但经过观察发现不符合实际抓包情况，故改成未知含义。

S2-0-0-1-0 Server Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Server Info Tag	0x0c01	0x0c01
2	Int16(le)	Server Info Length		0x000c
4	Int16(le)	RDP Version		0x0004
6	Int16(le)	Unknown	0x0008	0x0008
8	(Length－8)Byte	Reserved	0	0x00000000

表 22

表 22 中第一个字段是子数据包的 TAG 字段，可能的值为：0x0c01/Server Info，0x0c02/Server Encrypt Info，0x0c03/Server Channels Info；第二个字段的“Server Info Length”值等于整个子数据包的长度；最后一个字段值为全 0，长度等于(Length － 8)Byte。

S2-0-0-1-1 Server Channels Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Server Channels Info Tag	0x0c03	0x0c03
2	Int16(le)	Server Channels Info Length		0x0010
4	(Length－4)Byte	Unknown		0xeb0304...

表 23

表 23 和 22 类似，其中第三个字段内容是服务器通道信息，不过内部各值的含义未知，长度等于(Length - 4)Byte。

S2-0-0-1-2 Server Encrypt Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	Server Encrypt Info Tag	0x0c02	0x0c02
2	Int16(le)	Server Encrypt Info Length		0x00ec
4	Int32(le)	RC4 Key Size	0/None, 1/40bit, 2/128bit, 8/56bit,0x10/FIPS	0x00000002
8	Int32(le)	Encryption Level	0/None, 1/Low, 2/Client Compatible, 3/High, 4/FIPS	0x00000003
12	Int32(le)	Random Salt Len	0x20	0x00000020(32)
16	Int32(le)	RSA Info Len		0x000000b8(184)
20	32Byte	Server Salt		0xad57de...
52	Int32(le)	Encryption Type	1/RDP4-style, 0x80000002/X.509	0x00000001
56	Int32(le)	Unknown		0x00000001
60	Int32(le)	Unknown		0x00000001
64	Variable Length	Public Key Info		见 S2-0-0-1-2-0
160	Variable Length	Key Signature Info		见 S2-0-0-1-2-1

表 24

表 24 中“Server Encrypt Info Length”值为整个子数据包的长度（包括

它的子数据包)；从“Encryption Type”字段开始至该子数据包结束的这一部分，其实是一个子数据包，称为“RSA Info”，存储 RSA 加密的相关信息。“RSA Info Len”值也就等于这个“RSA Info”包的长度。

当“Encryption Type”值为 1 时，表示使用 RDP4 风格存储 RSA 信息，此时的字段结构如表 24 所示，当值为 0x80000002 时，表示使用 X.509 证书存储 RSA 信息，此时“RSA Info”部分的结构与表 24 不同，这种情况只在 Win2008 以上版本开启了身份验证时才出现，故在此不讨论。

当“RC4 Key Size”和“Encryption Level”值都为 0 时，表示此次 RDP 连接将不进行任何加密，此时该子数据包中“Encryption Level”后的所有字段将不存在（这种情况只有构造数据包才可能存在)。

S2-0-0-1-2-0 Public Key Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	BER_TAG_PUBLIC_KEY	0x0006	0x0006
2	Int16(le)	BER_TAG_LENGTH		0x005c
4	Int32(le)	RSA Magic	0x31415352	0x31415352
8	Int32(le)	RSA Modulus Len	64(256)/SEC_MODULUS_SIZE + 8/SEC_RESERVED_SIZE	0x00000048
12	8Byte	Unknown		0x00020000...
20	4Byte	RSA Exponent		0x01000100
24	(SEC_MODULUS_SIZE)Byte	RSA Modulus		0xd70b39...
88	8Byte	Reserved	0	0x0000...

表 25

表 25 中“RSA Modulus Len”的 SEC_MODULUS_SIZE 这个常数在 RDP5 版本（WinXP，Win2003）中等于 64，而在 RDP6 以上版本（Win2008，Win7）中则等于 256，它其实是指 RSA 公钥的长度。“RSA Exponent”和“RSA Modulus”都是公钥的组成部分，其中 Modulus 就是公钥值。

S2-0-0-1-2-1 Key Signature Info

Offset	Datatype	Description	Expect Value	Value
0	Int16(le)	BER_TAG_KEY_SIG	0x0008	0x0008
2	Int16(le)	BER_TAG_LENGTH		0x0048
4	(Length-8)Byte	Unknown		0x05cfb2...
68	8Byte	Reserved	0	0x0000...

表 26

表 26 与 25 类似，其中第三个字节的内容含义未知，一般认为是关于 Public Key 的签名信息，即可能是微软为了验证该公钥是否由服务器生成，以防止中间人攻击而设置的字段。

通道连接阶段数据包交换流程：客户端首先发送“建立域请求数据包”（MCS Erect Domain Request Packet）和“连接用户请求数据包”（MCS Attach User Request Packet）给服务器，服务器则回复“连接用户确认数据包”（MCS Attach User Confirm Packet）给客户端，其中包含用户 ID。然后客户端针对每个通道，发送“通道进入请求数据包”（MCS Channel Join Request Packet）给服务器，最后服务器对每个通道请求回复“通道进入确认数据包”（MCS Channel Join Confirm Packet），至此整个 MCS 层连接建立完毕（注意，客户端只有在上一个通道进入请求数据包被确认后，才会发送下一个通道请求）。

C3 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x000c(12)
4	Variable Length	TPDU		见 C3-0

表 27

C3-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)

2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Erect Domain Request		见 C3-0-0

表 28

C3-0-0 MCS Erect Domain Request Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Erect Domain Request Tag	0x04	0x04
1	Int16(be)	Height	0x0001	0x0001
3	Int16(be)	Interval	0x0001	0x0001

表 29

表 29 中第一个字段为标签字段，指示该子数据包的含义，其可能的取值为以下值左移 2 位后的数字（即乘以 4）：1/Erect_Domain_Request，8/Disconnect_Provider_Ultimatum，10/Attach_User_Request，11/Attach_User_Confirm，14/Channel_Join_Request，15/Channel_Join_Confirm,25/Send_Data_Request,26/Send_Data_Indication。后续几个通道连接阶段的数据包也都采用上述几个值。

C4 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0008(8)
4	Variable Length	TPDU		见 C4-0

表 30

C4-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Attach User Request		见 C4-0-0

表 31

C4-0-0 MCS Attach User Request Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Attach User Request Tag	0x28	0x28

表 32

表 30、31、32 与上一个数据包类似，不再做说明。

S3 TPKT from Server

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x000b(11)
4	Variable Length	TPDU		见 S3-0

表 33

S3-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Attach User Confirm		见 S3-0-0

表 34

S3-0-0 MCS Attach User Confirm Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Attach User Confirm Tag	0x2e	0x2e
1	Int8	Result Value	0	0x00
2	Int16(be)	User ID		0x0007

表 35

表 35 中标签字段的值只要右移 2 位后等于 11 即代表 Attach User Confirm

Packet；“Result Value”值为 0 表示连接用户成功；“User ID”值不唯一，若标签字段值的第 2 比特位为 0，则此字段值为 0，否则由服务器指定一个 ID 值，用于标识该客户端（User ID 值比较重要，后面的数据包会用到它）。

然后是建立多个通道连接阶段。该阶段中有几个通道就有几对通道进入数据包（MCS Channel Join Packet）。其中有两个通道是必须存在的，即“基本用户通道”（MCS_User_Channel_Base）和“全局通道”（MCS_Global_Channel）。而随后的几个通道则根据在“MCS Initial Packet”的“Client Channels Info”子数据包中的通道数量和标签号决定。

C5 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x000c(12)
4	Variable Length	TPDU		见 C5-0

表 36

C5-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Channel Join Request		见 C5-0-0

表 37

C5-0-0 MCS Channel Join Request Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Channel Join Request Tag	0x38	0x38
1	Int16(be)	User ID		0x0007
3	Int16(be)	Channel Flags		0x03f0

表 38

表 38 中“MCS Channel Join Request Tag”值等于 14/Channel_Join_Request 乘以 4，即 0x38；“User ID”字段要与“MCS Attach User Confirm Packet”中的“User ID”的值一致。

该通道是“基本用户通道”（MCS_User_Channel_Base）的请求包，该通道的标签值“Channel Flags”等于（User ID）+ 1001，其中 1001 是基本用户通道的初值。

S4 TPKT from Server

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x000f(15)
4	Variable Length	TPDU		见 S4-0

表 39

S4-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Channel Join Confirm		见 S4-0-0

表 40

S4-0-0 MCS Channel Join Confirm Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Channel Join Confirm Tag	0x3e	0x3e
1	Int8	Result Value	0	0x00
2	Int16(be)	User ID		0x0007
4	Int16(be)	Channel Flags		0x03f0
6	Int16(be)	Channel Flags Confirmed		0x03f0

表 41

表 39-41 的数据包是对“基本用户通道”（MCS_User_Channel_Base）的确认包。其中表 41 的标签字段值只要右移 2 位后等于 15 即代表 Channel Join Confirm Packet；“Result Value”值为 0 表示确认成功。

“User ID”和“Channel Flags”的值必须与通道请求包中对应字段的值相等；“Channel Flags Confirmed”字段表示服务器实际为该通道标签赋的值，一般就等于客户端请求包中的值。

随后两个数据包（C6，S5）是“全局通道”的请求和确认包，其格式基本与（C5，S4）相同，要注意的是该通道的“Channel Flags”值为 0x03eb(1003)。该通道一般认为是服务器传输图像数据的通道。

然后就是剩下几个通道的请求和确认了，具体数目和“Channel Flags”值参见“MCS Initial Packet”中“Client Channels Info”子数据包，格式与（C5，S4）也基本相同，在此不多做介绍。在我们实际抓包中，此处有 4 个通道，其数据包分别是：（C7，S6），（C8，S7），（C9，S8），（C10，S9）。

3. SEC 层连接

该层主要功能是客户端与服务器交换加密解密，证书等安全信息。其大致流程如下：客户端发送“Security Exchange Packet”和“Client Info Packet”；服务器发送“License Error Packet - Valid Client”和“Demand Active Packet”；客户端发送“Confirm Active Packet”。

至此，SEC 层连接建立完毕。具体数据包字段格式如下：

C11 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x005e(94)
4	Variable Length	TPDU		见 C11-0

表 42

C11-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)

1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Send Data Request		**见 C11-0-0**

表 43

C11-0-0 MCS Send Data Request Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Send Data Request Tag	0x64	0x64
1	Int16(be)	User ID		0x0007
3	Int16(be)	Channel Flags		0x03eb
5	Int8	Flags	0x70	0x70
6	Int8/Int16(be)	MCS Data Length	Length/ (0x8000|Length)	0x50
7	Variable Length	Security Exchange Packet		**见 C11-0-0-0**

表 44

表 44 中标签字段的值等于 25/Send_Data_Request 乘以 4,即 0x64；“User ID”值与前面数据包的 User ID 一致；“Channel Flags”等于“全局通道”（Global Channel）的标签值；

“MCS Data Length”值表示从下一个字段开始到子数据包结束的长度，但它的格式有两种情况：在 RDP5 版本（WinXP，Win2003）中，该字段用 Int8 表示，值就等于长度；RDP6 以上版本（Win2008，Win7）中，则用 Int16(be)表示，且值等于（0x8000 | Length），其中 Length 为实际长度。

C11-0-0-0 Security Exchange Packet

Offset	Datatype	Description	Expect Value	Value
0	Int32(be)	Unknown	0x01020000	0x01020000
4	Int32(le)	Sec Data Length		0x00000048
8	(SEC_MODULUS_SIZE) Byte	Encrypted Client Random		0x06055c...

72	8Byte	Reserved	0	0x0000...

表 45

表 45 中“Sec Data Length”值等于下一个字段开始直至该子数据包结束的长度（其实就是 SEC 层数据部分长度）；“Encrypted Client Random”为使用 RSA 公钥加密后的客户端随机数值，它的长度等于常数 SEC_MODULUS_SIZE，该常数在 RDP5 中等于 64，RDP6 以上版本等于 256。

“Client Info Packet”如下所示：

C12 TPKT from Client

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPKT Version	3	0x03(3)
1	Int8	TPKT Reserved	0	0x00
2	Int16(be)	TPKT Length		0x0189(393)
4	Variable Length	TPDU		见 C12-0

表 46

C12-0 TPDU

Offset	Datatype	Description	Expect Value	Value
0	Int8	TPDU Length		0x02(2)
1	Int8	TPDU Packet Type	0xf0	0xf0(240)
2	Int8	TPDU eot	0x80	0x80(128)
3	Variable Length	MCS Send Data Request		见 C12-0-0

表 47

C12-0-0 MCS Send Data Request Packet

Offset	Datatype	Description	Expect Value	Value
0	Int8	MCS Send Data Request Tag	0x64	0x64
1	Int16(be)	User ID		0x0007
3	Int16(be)	Channel Flags		0x03eb
5	Int8	Flags	0x70	0x70
6	Int16(be)	MCS Data Length	(0x8000|Length)	0x817a

8	Variable Length	Security Exchange Packet		见 C12-0-0-0

表 48

表 48 与 44 类似，其中“MCS Data Length”固定为 Int16(be)表示，值等于（0x8000|Length）；

C12-0-0-0 Client Info Packet

Offset	Datatype	Description	Expect Value	Value

表 49

从表 49 开始，所有数据都经过 RC4 加密，故后续数据包真实字段信息暂不可知，需要进一步调研。待所有数据包解密以后，再将字段格式分析公布。

更多信息请访问作者主页：Http://keary.cn/

2012.5.22