Sécurité de *Remote Desktop Protocol*

Aurélien Bordes, Arnaud Ebalard,
Raphaël Rigo
prenom.nom@ssi.gouv.fr

SSTIC 2012

Agence Nationale de la Sécurité
des Systèmes d'Information
51, boulevard de la Tour-Maubourg
75700 Paris 07 SP

Contexte

Solutions d'administration à distance sous Windows :

- VNC : historique, peu sécurisé
- RDP
- RPC : via la *MMC*, limité aux produits MS
- solutions propriétaires

Critères de choix :

- l'environnement de déploiement
- les fonctionnalités
- la sécurité (accessoirement)

Limites de la présentation

Évolutions :

- RDP est initialement un protocole de déport d'affichage ...
- ...étendu avec de nouvelles fonctionnalités au fil des versions ...
- ...maintenant au cœur d'une architecture de services avec 2008 R2.

Couverture de l'étude :

- La sécurité du cœur du protocole RDP

Hors étude :

- La sécurité des architectures RDS (*Remote Desktop Services*)
- La sécurité intrinsèque des fonctionnalités et extensions

Windows Server 2008 R2: Remote Desktop Services Component Architecture

Vous êtes ici!

Acronyms

AD DS
Active Directory Domain Services
APP-V
Application Virtualization
CAL
client access license
GPU
graphics processing unit
MSI
Microsoft Installer Package
RDC
Remote Desktop Connection
RDS
Remote Desktop Services
RDP
Remote Desktop Protocol
RD Connection Broker
Remote Desktop Connection Broker
RD Gateway
Remote Desktop Gateway
RD Session Host
Remote Desktop Session Host
RD Virtualization Host
Remote Desktop Virtualization Host
RD Web Access
Remote Desktop Web Access
ROI
Return On Investment
SP1
Service Pack 1
SCVMM
System Center Virtual Machine Manager
SSE2
Streaming SIMD Extensions 2
SLAT
Second-Level Address Translation
SSL
Secure Sockets Layer
VDI
Virtual Desktop Infrastructure
V-GPU
virtual graphics processing unit
VMM
Virtual Machine Manager

Windows Server 2008 R2
Remote Desktop Services

Remote D

An RD Session Host server ho
Desktop Services clients. Use
and to use network resource
Connection Broker, RD Web A

RD Session Host server farms
You can use a load-balanced RD Session Host server farm to scale the performance of a single RD Session Host server by distributing Remote Desktop Services sessions across multiple servers.

Deliver

Session-based desktops delivery
An RD Session Host server can deliver a user desktop session to any designated user in the network.

RD Session Host configuration
1. Configure Network Level Authentication for the RD Session Host server
2. Configure the Remote Desktop Users group to give users and groups permissions to remotely connect
3. Configure an RD Session Host server running Windows Server 2008 R2 to use at least one Remote Desktop license server

Remote Desktop Connection client configuration
The RD Session Host server can be configured so that users connecting to a remote session can use some of the following functionality:
1. Audio recording redirection
2. Windows Aero experience
3. Remote computer's audio and video playback redirection

RemoteApp programs delivery
RemoteApp enables administrators to make programs that are accessed remotely through an RD Session Host server appear as if they are running on the client computer. Instead of being presented to the user in the desktop of the RD Session Host server, the RemoteApp program is integrated with the client computer.
1. Install the RD Session Host role service on the designated server before installing any applications that will need to run as RemoteApp programs.
2. Install related applications, or applications that have dependencies on other local applications, on the same RD Session Host server.
3. Deploy applications on separate RD Session Host servers in the following scenarios:
   1. Applications have compatibility issues that affect other programs.
   2. Resource intensive applications that could affect performance when running multiple instances on the same server.

Accessing RemoteApp programs
Users access RemoteApp programs in the following ways:
1. Remote Desktop Protocol (.rdp) file
2. RD Web Access Web portal
3. RemoteApp And Desktop Connection by using the Windows 7 Start menu

Client requirements for accessing RemoteApp programs
1. To access RemoteApp programs, the client computer must be running at least RDC 6.1.
2. To access RemoteApp programs, the client computer must be running either Windows 7, Windows Vista with SP1, or Windows XP with SP3.
3. To access RemoteApp and Desktop Connection through the Start menu requires Windows 7.

Remote Desktop Connection client

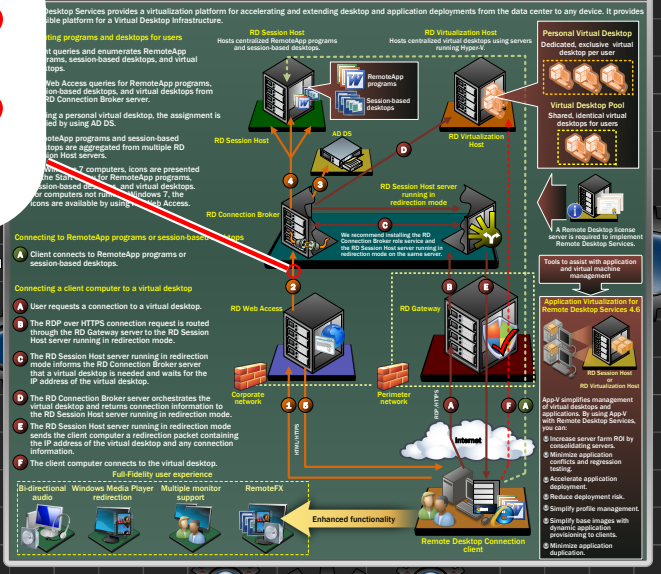
Remote Desktop Services Architecture

Desktop Services provides a virtualization platform for accelerating and extending desktop and application deployments from the data center to any device. It provides
ible platform for a Virtual Desktop Infrastructure.

ing programs and desktops for users
t queries and enumerates RemoteApp
rams, session-based desktops, and virtual
tops.
Web Access queries for RemoteApp programs,
on-based desktops, and virtual desktops from
RD Connection Broker server.
ing a personal virtual desktop, the assignment is
ed by using AD DS.
oteApp programs and session-based
ktops are aggregated from multiple RD
on Host servers.
7 computers, icons are presented
the Start menu for RemoteApp programs,
ession-based desktops, and virtual desktops.
r computers not running Windows 7, the
icons are available by using RD Web Access.

Connecting to RemoteApp programs or session-based desktops
A. Client connects to RemoteApp programs or session-based desktops.

Connecting a client computer to a virtual desktop
A. User requests a connection to a virtual desktop.
B. The RDP over HTTPS connection request is routed through the RD Gateway server to the RD Session Host server running in redirection mode.
C. The RD Session Host server running in redirection mode informs the RD Connection Broker server that a virtual desktop is needed and waits for the IP address of the virtual desktop.
D. The RD Connection Broker server orchestrates the virtual desktop and returns connection information to the RD Session Host server running in redirection mode.
E. The RD Session Host server running in redirection mode sends the client computer a redirection packet containing the IP address of the virtual desktop and any connection information.
F. The client computer connects to the virtual desktop.

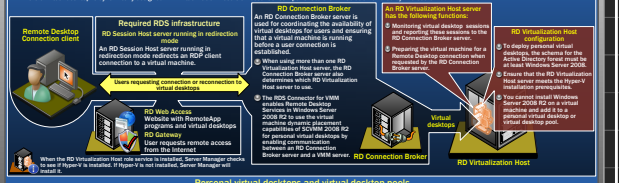


Full-Fidelity user experience
Bi-directional audio | Windows Media Player redirection | Multiple monitor support | RemoteFX
Enhanced functionality

RDS

Remote Desktop Virtualization Host

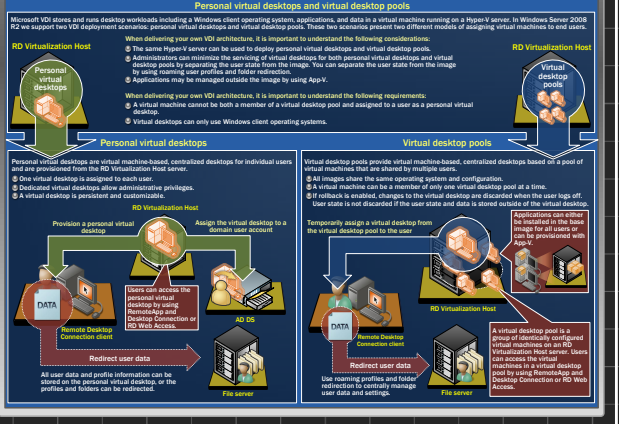
Understanding Remote Desktop Virtualization Host
Remote Desktop Virtualization Host is a new Remote Desktop Services role service included with Windows Server 2008 R2. RD Virtualization Host integrates with Hyper-V to provide virtual machines that can be used as personal virtual desktops or virtual desktop pools. User accounts can be assigned a unique personal virtual desktop or be redirected to a virtual desktop pool where a virtual desktop is dynamically assigned. RD Virtualization Host is an essential component to the VDI solution offered by Microsoft.



Personal virtual desktops and virtual desktop pools
Microsoft VDI stores and runs desktop workloads including a Windows client operating system, applications, and data in a virtual machine running on a Hyper-V server. In Windows Server 2008 R2 we support two VDI deployment scenarios: personal virtual desktops and virtual desktop pools. These two scenarios present two different models of assigning virtual machines to end users.

When delivering your own VDI architecture, it is important to understand the following considerations:
1. The same Hyper-V server can be used to deploy personal virtual desktops and virtual desktop pools.
2. Administrators can minimize the servicing of virtual desktops for both personal virtual desktops and virtual desktop pools by separating the user state from the image. You can separate the user state from the image by using roaming user profiles and folder redirection.
3. Applications may be managed outside the image by using App-V.

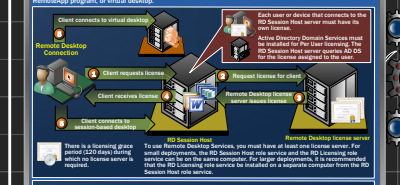
When delivering your own VDI architecture, it is important to understand the following requirements:
1. A virtual machine cannot be both a member of a virtual desktop pool and assigned to a user as a personal virtual desktop.
2. Virtual desktops can only use Windows client operating systems.



Remote Desktop Licensing

Remote Desktop Licensing manages the Remote Desktop Services client access licenses and VDI Suite licenses that are required for each device or user to connect to session-based desktops, RemoteApp programs, or virtual desktops.

When a client connects to an RD Session Host server, the RD Session Host server determines if a license is needed. The RD Session Host server then requests an RDS CAL or VDI Suite license from a Remote Desktop license server on behalf of the client. If an appropriate license is available from a license server, the RDS CAL or VDI Suite license is issued to the client, and the client will be able to connect to the session-based desktop, RemoteApp program, or virtual desktop.



There is a licensing grace period (120 days) during which no license server is required.

To use Remote Desktop Services, you must have at least one license server. For small deployments, the RD Session Host role service and the RD Licensing role service can be on the same computer. For larger deployments, it is recommended that the RD Licensing role service be installed on a separate computer from the RD Session Host role service.

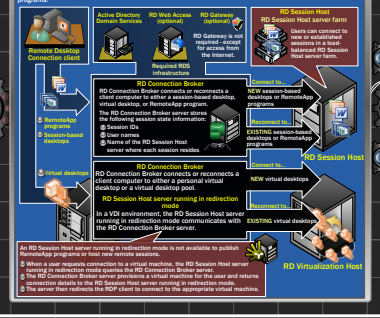
Microsoft Virtual Desktop Infrastructure Licensing
For VDI implementations, the following options are available:

VDI Standard Suite
The VDI Standard Suite is for deploying the basic infrastructure for VDI. Hyper-V Server 2008 R2 provides the virtualization platform; management of the virtual infrastructure is provided by system center components.

VDI Premium Suite
The VDI Premium Suite is for customers wanting additional flexibility from their VDI environment. It includes all of the VDI Standard Suite options, including unrestricted Remote Desktop Services, enabling customers to deliver both session-based desktops and virtual desktops.

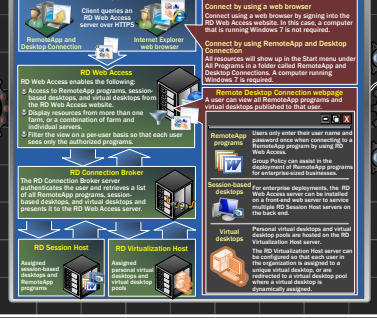
Remote Desktop Connection Broker

Remote Desktop Connection Broker provides a single, personalized, and aggregated view of RemoteApp programs, session-based desktops, and virtual desktops to users. RD Connection Broker supports load balancing and reconnection to existing sessions on virtual desktops, session-based desktops, and RemoteApp programs.



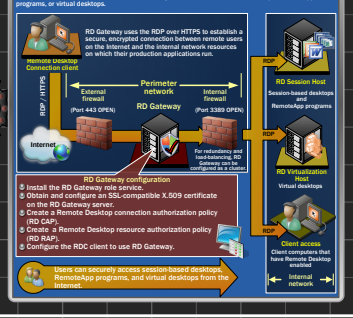
Remote Desktop Web Access

A Remote Desktop Web Access server provides users with a customizable web portal for accessing session-based desktops, virtual desktops, and RemoteApp programs.



Remote Desktop Gateway

The Remote Desktop Gateway role service in Windows Server 2008 R2 enables compatible devices to securely connect over the Internet to RD Session Host servers or RD Virtualization Host servers behind a corporate firewall. Network resources can be any authorized session-based desktops, RemoteApp programs, or virtual desktops.



MIcrosoft RemoteFX

Microsoft RemoteFX delivers a rich user experience for VDI by using graphics processing units that are present on the server and shared across multiple virtual desktops.

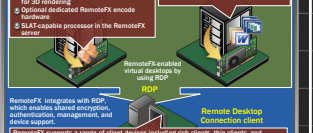
RemoteFX contains the following elements:
1. Host-side V-GPU rendering and capture for 3D
2. Improved encode/decode within RDP
3. Generic USB redirection
4. Enablement of ultra-thin client types

RemoteFX for RD Virtualization Host
RemoteFX enables:
1. Rendered rich 3D content by using a virtualized GPU with Windows 7 Enterprise with SP1 or Windows 7 Ultimate with SP1
2. Generic USB redirection
3. Improved encode/decode that improves efficiency and user experience
4. Ultra-thin client support

RemoteFX support requirements:
1. GPU in the RD Virtualization Host server for 3D rendering
2. Optional dedicated RemoteFX encode hardware
3. SLAT-capable processor in the RemoteFX server

RemoteFX for RD Session Host
RemoteFX enables:
1. Improved encode/decode that improves efficiency and user experience
2. Ultra-thin client support

RemoteFX support requirements:
1. Optional dedicated RemoteFX encode hardware
2. SSE2-capable processor in the RD Session Host server



RemoteFX supports a range of client devices including rich clients, thin clients, and ultra-thin clients.
1. The software client is part of Remote Desktop Connection that is included with Windows client and server operating systems and Windows Embedded.
2. An RDP client with RemoteFX can run as software or by using a RemoteFX hardware decoder.

Author: Martin McClean with contributions from Brian Lich and Thomas Willingham (Microsoft Windows Server Information Experience Team, Redmond WA. email: rdsua@microsoft.com)

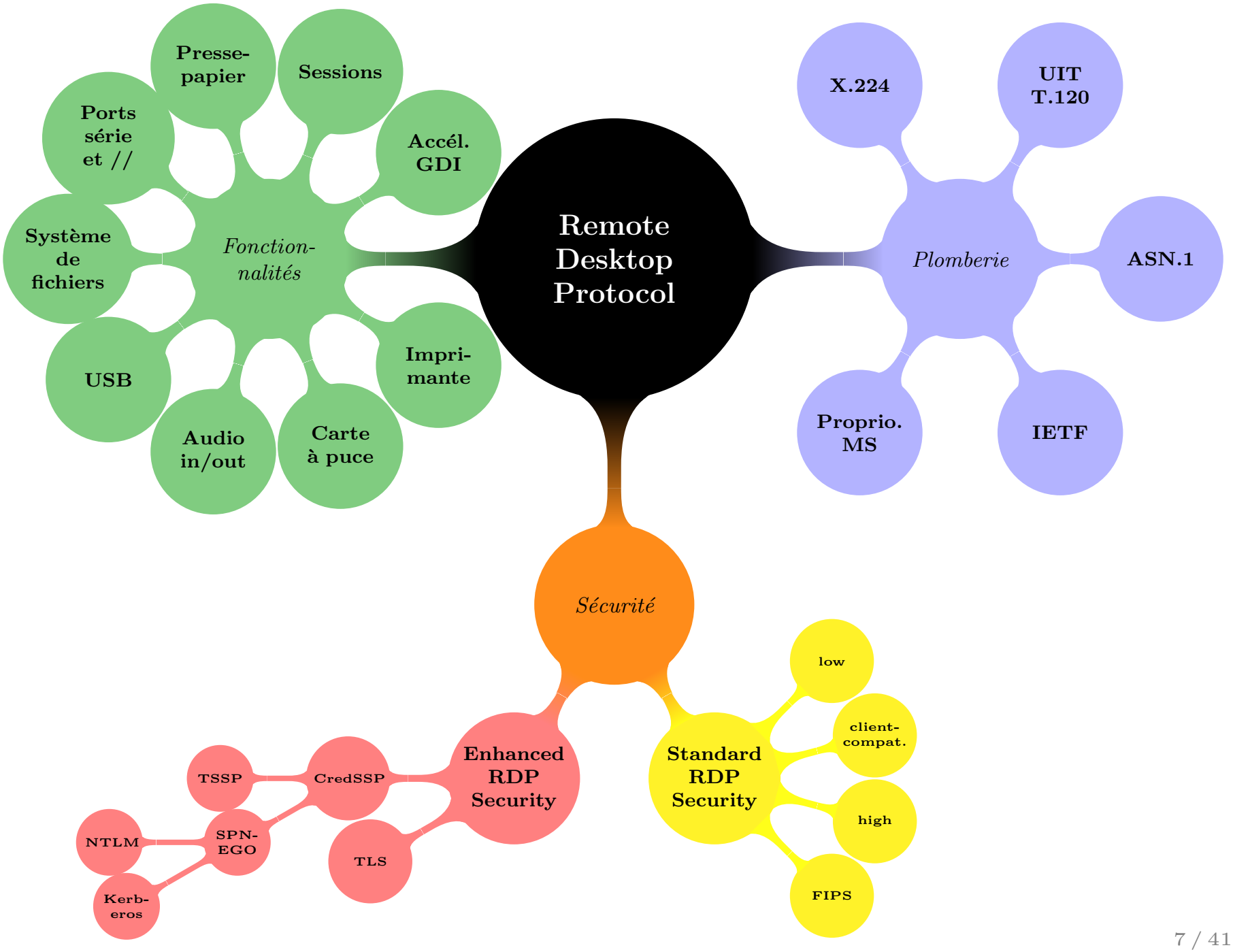

Remote Desktop Protocol
Fonction-nalités
Presse-papier
Sessions
Ports série et //
Accél. GDI
Système de fichiers
Impri-mante
USB
Audio in/out
Carte à puce
Plomberie
X.224
UIT T.120
ASN.1
Proprio. MS
IETF
Sécurité
Enhanced RDP Security
TSSP
CredSSP
NTLM
SPN-EGO
Kerb-eros
TLS
Standard RDP Security
low
client-compat.
high
FIPS

Table des matières

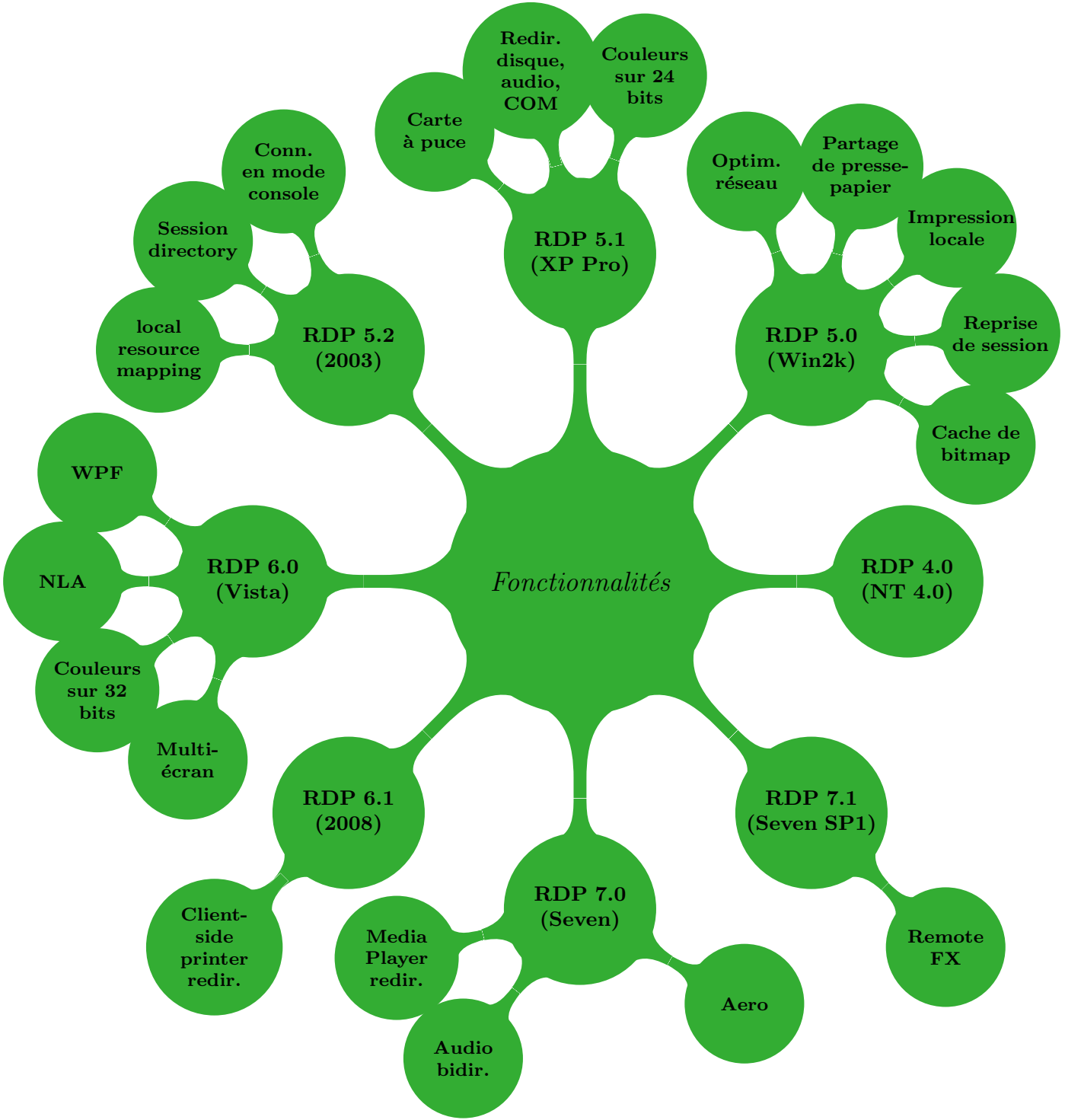
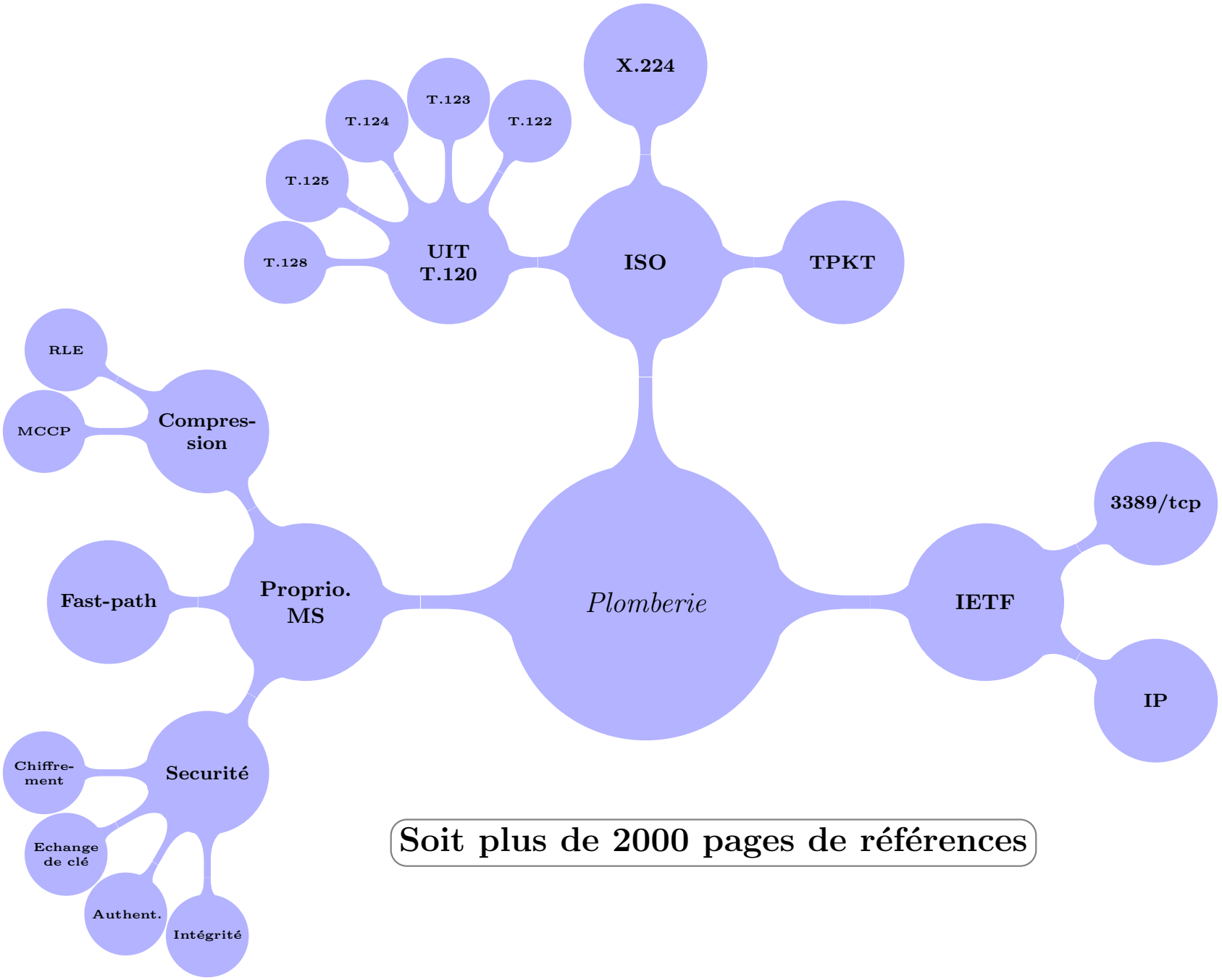

Fonctionnalités
RDP 5.1 (XP Pro)
Redir. disque, audio, COM
Couleurs sur 24 bits
Carte à puce
RDP 5.0 (Win2k)
Optim. réseau
Partage de presse-papier
Impression locale
Reprise de session
Cache de bitmap
RDP 4.0 (NT 4.0)
RDP 5.2 (2003)
Conn. en mode console
Session directory
local resource mapping
RDP 6.0 (Vista)
WPF
NLA
Couleurs sur 32 bits
Multi-écran
RDP 6.1 (2008)
Client-side printer redir.
RDP 7.0 (Seven)
Media Player redir.
Audio bidir.
Aero
RDP 7.1 (Seven SP1)
Remote FX

Table des matières



Soit plus de 2000 pages de références

```
Frame: Number = 10, Captured Frame Length = 338, MediaType = ETHERNET
Ethernet: Etype = Internet IP (IPv4),DestinationAddress:[08-00-27-B2-85-20],SourceAddress:[08-00-27-6E-4D-33]
Ipv4: Src = 192.168.0.2, Dest = 192.168.0.1, Next Protocol = TCP, Packet ID = 555, Total IP Length = 324
Tcp: Flags=...AP..., SrcPort=1031, DstPort=MS WBT Server(3389), PayloadLen=284, Seq=2644391141 - 2644391425,
ISOTS: TPKTCount = 1
TPKT: version: 3, Length: 284
  version: 3 (0x3)
  Reserved: 0 (0x0)
  PacketLength: 284 (0x11C)
X224: Data
  Length: 2 (0x2)
  Type: Data
  EOT: 128 (0x80)
T125: MCSConnect Initial
  MCSConnectInitial: Identifier=Generic Conference Contro (0.0.20.124.0.1), ConnectPDULength=166
    ConnectInitialHeader:
      AsnId: Application Constructed Tag (101)
        HighTag:
          Class:     (01......) Application (1)
          Type:      (..1.....) Constructed
          TagNumber: (...11111)
          TagValueEnd: 101 (0x65)
      AsnLen: Length = 272, LengthOfLength = 2
        LengthType: LengthOfLength = 2
        Length: 272 bytes
    CallingDomainSelector: 0x1
      AsnOctetStringHeader:
        AsnId: OctetString type (Universal 4)
          LowTag:
            Class:     (00......) Universal (0)
            Type:      (..0.....) Primitive
            TagValue: (...00100) 4
        AsnLen: Length = 1, LengthOfLength = 0
          Length: 1 bytes, LengthOfLength = 0
      OctetStream: 0x1
    CalledDomainSelector: 0x1
      AsnOctetStringHeader:
        AsnId: OctetString type (Universal 4)
          LowTag:
            Class:     (00......) Universal (0)
```

```
Frame: Number = 10, Captured Frame Length = 338, MediaType = ETHERNET
Ethernet: Etype = Internet IP (IPv4),DestinationAddress:[08-00-27-B2-85-20],SourceAddress:[08-00-27-6E-4D-33]
Ipv4: Src = 192.168.0.2, Dest = 192.168.0.1, Next Protocol = TCP, Packet ID = 555, Total IP Length = 324
Tcp: Flags=...AP..., SrcPort=1031, DstPort=MS WBT Server(3389), PayloadLen=284, Seq=2644391141 - 2644391425,
ISOTS: TPKTCount = 1
TPKT: version: 3, Length: 284
  version: 3 (0x3)
  Reserved: 0 (0x0)
  PacketLength: 284 (0x11C)
X224: Data
  Length: 2 (0x2)
  Type: Data
  EOT: 128 (0x80)
T125: MCSConnect Initial
MCSConnectInitial: Identifier=Generic Conference Contro (0.0.20.124.0.1), ConnectPDULength=166
  ConnectInitialHeader:
    AsnId: Application Constructed Tag (101)
      HighTag:
        Class:     (01......) Application (1)
        Type:      (..1.....) Constructed
        TagNumber: (...11111)
        TagValueEnd: 101 (0x65)
    AsnLen: Length = 272, LengthOfLength = 2
      LengthType: LengthOfLength = 2
      Length: 272 bytes
  CallingDomainSelector: 0x1
    AsnOctetStringHeader:
      AsnId: OctetString type (Universal 4)
        LowTag:
          Class:     (00......) Universal (0)
          Type:      (..0.....) Primitive
          TagValue: (...00100) 4
      AsnLen: Length = 1, LengthOfLength = 0
        Length: 1 bytes, LengthOfLength = 0
    OctetStream: 0x1
  CalledDomainSelector: 0x1
    AsnOctetStringHeader:
      AsnId: OctetString type (Universal 4)
        LowTag:
          Class:     (00......) Universal (0)
          Type:      (..0.....) Primitive
          TagValue: (...00100) 4
      AsnLen: Length = 1, LengthOfLength = 0
        Length: 1 bytes, LengthOfLength = 0
    OctetStream: 0x1
  UpwardFlag: True
    AsnBooleanHeader:
      AsnId: Boolean type (Universal 1)
        LowTag:
          Class:     (00......) Universal (0)
          Type:      (..0.....) Primitive
          TagValue: (...00001) 1
      AsnLen: Length = 1, LengthOfLength = 0
        Length: 1 bytes, LengthOfLength = 0
    Tf: 255 (0xFF)
  TargetParameters: Length = 25, LengthOfLength = 0
    DomainParametersHeader: 0x1
      AsnId: Sequence and SequenceOf types (Universal 16)
        LowTag:
          Class:     (00......) Universal (0)
          Type:      (..1.....) Constructed
          TagValue: (...10000) 16
      AsnLen: Length = 25, LengthOfLength = 0
        Length: 25 bytes, LengthOfLength = 0
      ChannelIDs: 34
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 34 (0x22)
      UserIDs: 2
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 2 (0x2)
      TokenIDs: 0
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 0 (0x0)
      NumPriorities: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MinThroughput: 0
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 0 (0x0)
      Height: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MCSPDUsize: 65535
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 65535 (0xFFFF)
      protocolVersion: 2
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 2 (0x2)
    MinimumParameters: Length = 25, LengthOfLength = 0
      DomainParametersHeader: 0x1
        AsnId: Sequence and SequenceOf types (Universal 16)
          LowTag:
            Class:     (00......) Universal (0)
            Type:      (..1.....) Constructed
            TagValue: (...10000) 16
        AsnLen: Length = 25, LengthOfLength = 0
          Length: 25 bytes, LengthOfLength = 0
      ChannelIDs: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      UserIDs: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      TokenIDs: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      NumPriorities: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MinThroughput: 0
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 0 (0x0)
      Height: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MCSPDUsize: 1056
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 1056 (0x420)
      protocolVersion: 2
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 2 (0x2)
    MaximumParameters: Length = 28, LengthOfLength = 0
      DomainParametersHeader: 0x1
        AsnId: Sequence and SequenceOf types (Universal 16)
          LowTag:
            Class:     (00......) Universal (0)
            Type:      (..1.....) Constructed
            TagValue: (...10000) 16
        AsnLen: Length = 28, LengthOfLength = 0
          Length: 28 bytes, LengthOfLength = 0
      ChannelIDs: 65535
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 65535 (0xFFFF)
      UserIDs: 64535
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 64535 (0xFC17)
      TokenIDs: 65535
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 65535 (0xFFFF)
      NumPriorities: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MinThroughput: 0
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 0 (0x0)
      Height: 1
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 1 (0x1)
      MCSPDUsize: 65535
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 2, LengthOfLength = 0
            Length: 2 bytes, LengthOfLength = 0
        AsnInt: 65535 (0xFFFF)
      protocolVersion: 2
        AsnIntegerHeader:
          AsnId: Integer type (Universal 2)
            LowTag:
              Class:     (00......) Universal (0)
              Type:      (..0.....) Primitive
              TagValue: (...00010) 2
          AsnLen: Length = 1, LengthOfLength = 0
            Length: 1 bytes, LengthOfLength = 0
        AsnInt: 2 (0x2)
  UserData: Identifier=Generic Conference Contro (0.0.20.124.0.1), ConnectPDULength=166
    UserDataHeader:
      AsnId: OctetString type (Universal 4)
        LowTag:
          Class:     (00......) Universal (0)
          Type:      (..0.....) Primitive
          TagValue: (...00100) 4
      AsnLen: Length = 175, LengthOfLength = 2
        LengthType: LengthOfLength = 2
        Length: 175 bytes
    AsnBerObjectIdentifier: Generic Conference Contro (0.0.20.124.0.1)
      AsnObjectIdentifierHeader:
        AsnId: Reserved for use by the encoding rules (Universal 0)
          LowTag:
            Class:     (00......) Universal (0)
            Type:      (..0.....) Primitive
            TagValue: (...00000) 0
      AsnLen: Length = 5, LengthOfLength = 0
        Length: 5 bytes, LengthOfLength = 0
    First: 0 (0x0)
    Final: 20 (0x14)
    Final: 124 (0x7C)
    Final: 0 (0x0)
    Final: 1 (0x1)
  ConnectPDULength: 166
    Align: No Padding
    Length: 166
  ConnectGCCPDU: conferenceCreateRequest
    ExtensionBit: 0 (0x0)
    ChoiceValue: conferenceCreateRequest
      Value: (000.....) 0x0
    conferenceCreateRequest:
      ExtensionBit: 0 (0x0)
      convenerPasswordPresent: 0 (0x0)
      passwordPresent: 0 (0x0)
      conductorPrivilegesPresent: 0 (0x0)
      conductedPrivilegesPresent: 0 (0x0)
      nonConductedPrivilegesPresent: 0 (0x0)
      conferenceDescriptionPresent: 0 (0x0)
      callerIdentifierPresent: 0 (0x0)
      userDataPresent: 1 (0x1)
      conferenceName:
        ExtensionBit: 0 (0x0)
        textPresent: 0 (0x0)
        numeric: 1
          SimpleNumericString: 1
            NumericString: 1
              Align: No Padding
                Padding1: (0.......) 0x0
              Length: 1
                Value: (00000000) 0x0
              Restrictedstr: 1
                FourBits: (0001....) 0x1
      lockedConference: False
        Value: False 0.......
      listedConference: False
        Value: False 0.......
      conductibleConference: False
        Value: False 0.......
      TerminationMethod: automatic
        ExtensionBit: 0 (0x0)
        RootIndex: 0
          Value: (0.......) 0x0
      userData:
        Size: 1
          Align: No Padding
            Padding7: (0000000.) 0x0
          Length: 1
        UserData: 0x44756361
          valuePresent: 1 (0x1)
          key: h221NonStandard
            ChoiceValue: h221NonStandard
              Value: (1.......) 0x1
            h221NonStandard:
              H221NonStandardIdentifier: length: 4
                ConstrainedLength: 4
                  Value: (00000000) 0x0
                Align: No Padding
                  Padding6: (000000..) 0x0
                Value: Binary Large Object (4 Bytes)
            ClientMcsConnectInitialPdu:
              RDPGCCUserDataRequestLength: 152
                Align: No Padding
                Length: 152
              TsUd: CS_CORE
                TsUdHeader: Type = CS_CORE, Length = 140
                  Type: CS_CORE
                  Length: 140 (0x8C)
                TsUdCsCore:
                  Version: RDP 5.0, 5.1, 5.2, 6.0, 6.1, and 7.0
                  DesktopWidth: 800 (0x320)
                  DesktopHeight: 600 (0x258)
                  ColorDepth: 8 bpp
                  SASSequence: 0xaa03, SHOULD be set to RNS_UD_SAS_DEL(0xAA03)
                  KeyboardLayout: Language: French, Location: France
                  ClientBuild: 2195 (0x893)
                  ClientName: AWESOME
                  KeyboardType: IBM enhanced (101- or 102-key) keyboard
                  KeyboardSubType: 0 (0x0)
                  KeyboardFunctionKey: 12 (0xC)
                  ImeFileName:
                  PostBeta2ColorDepth: 8 bpp
                  ClientProductId: 0x1, SHOULD be set to initialized to 1
                  SerialNumber: 0x0, SHOULD be set to 0
              TsUd: CS_SECURITY
                TsUdHeader: Type = CS_SECURITY, Length = 12
                  Type: CS_SECURITY
                  Length: 12 (0xC)
                TsUdCsSec:
                  EncryptionMethod:
                    Support40Bit:  (.............................0) Not Support
                    Support128Bit: (.............................0.) Not Support 128-bit
                    Reserved1:     (............................0..)
                    Support56Bit:  (...........................0...) Not Support 56-bit
                    SupportFIPS:   (..........................0....) Not Support FIPS Compliant
                    Reserved2:     (000000000000000000000000000.....)
                  ExtEncryptionMethod: Support 40-bit;Support 56-bit;
                    Support40Bit:  (.............................1) Support 40-bit
                    Support128Bit: (.............................0.) Not Support 128-bit
                    Reserved1:     (............................0..)
                    Support56Bit:  (...........................1...) Support 56-bit
                    SupportFIPS:   (..........................0....) Not Support FIPS Compliant
                    Reserved2:     (000000000000000000000000000.....)
```

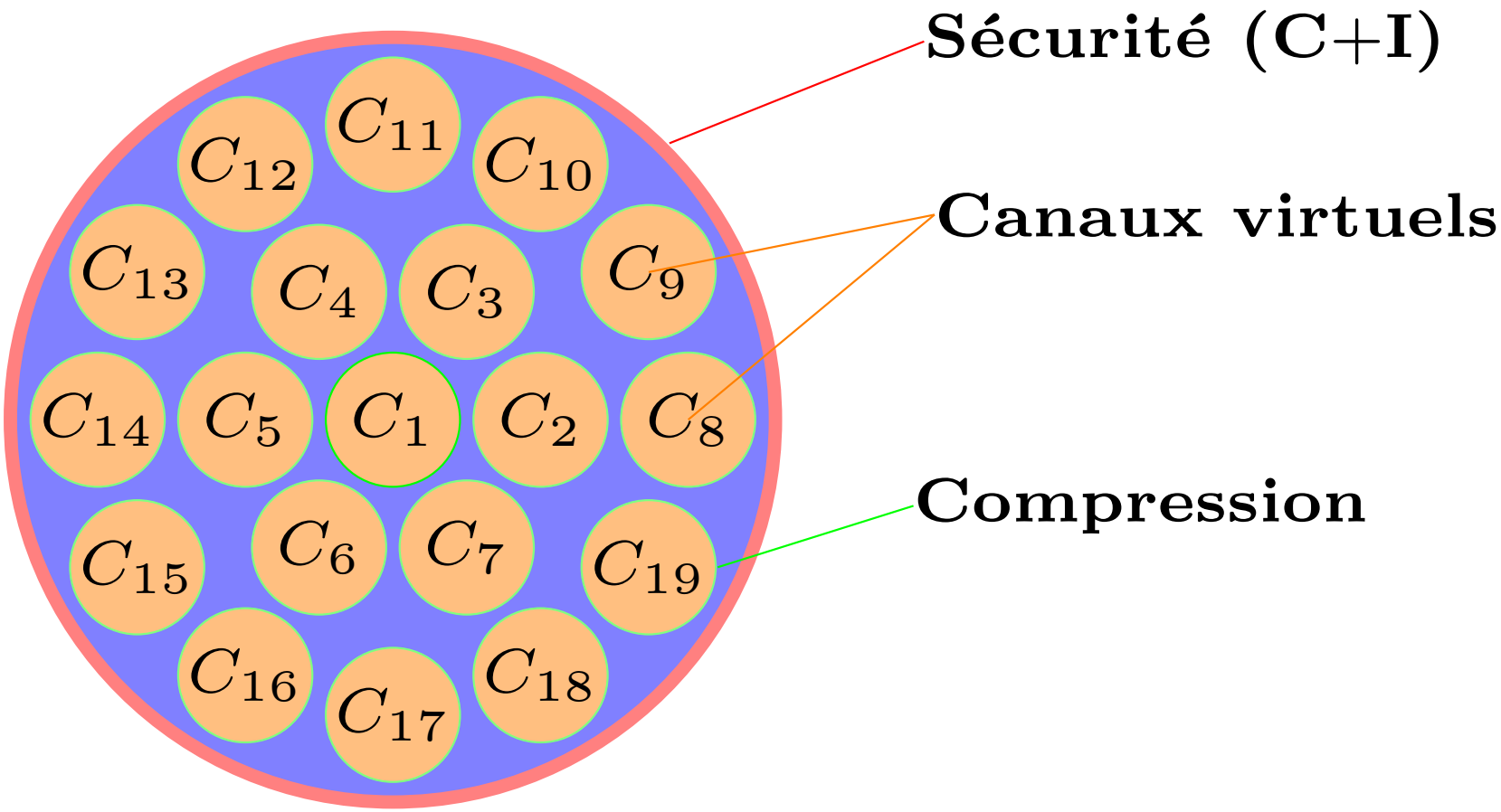
Architecture protocolaire

Panorama :

- RDP procède à la mise en place de canaux de communication (*virtual channels*) entre composants (matériel ou logiciel) des systèmes locaux et distants.
- Ces canaux permettent l'échange de données :
  - Entrées utilisateur clavier/souris (*input events*) : keycodes, etc. ;
  - retour graphiques (*output events*) : bitmap, glyphes, etc. ;
  - copier/coller par le presse-papier, etc. ;
  - montage de système de fichiers ;
  - son ;
  - ...

[Complexité d'une] montée de session RDP :

- 8 étapes distinctes ;
- plusieurs dizaines de paquets échangés ;
- des centaines de paramètres négociés.


C_{11}
C_{12}
C_{10}
C_{13}
C_4
C_3
C_9
C_{14}
C_5
C_1
C_2
C_8
C_{15}
C_6
C_7
C_{19}
C_{16}
C_{17}
C_{18}
Sécurité (C+I)
Canaux virtuels
Compression

Couches de sécurité

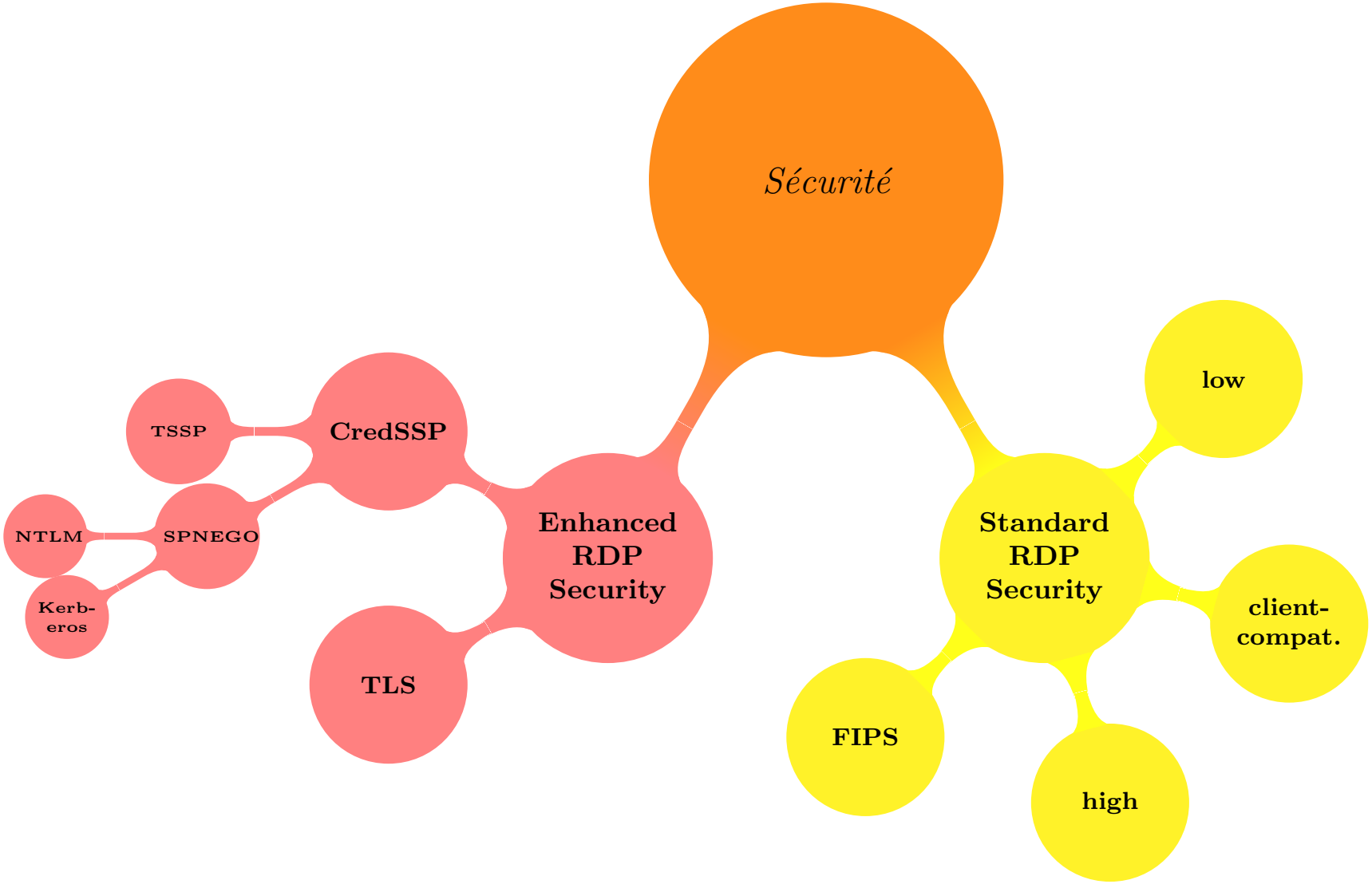


Table des matières

Interlude récréatif #1

- Contexte :
  - client Windows Seven à jour vers un Windows 2003 à jour ;
  - capture passive de la session.
- Pré-requis attaquant :
  - factorisation de la clé (quelques jours) ou
  - compromission serveur.
- Problèmes sous-jacents :
  - taille de clé extrêmement faible (512 bits) ;
  - schéma d'échange de clé sans PFS.
- Amélioration : passage à des clés de 2048 bits avec Windows 2008.

Vidéo

Mécanismes *Standard Security* (1/2)

- **Échange de clé**
  - aléa du client chiffré par la clé publique du serveur :
    - clé de 512 bits jusqu'à Windows 2003,
    - 2048 bits depuis Windows 2008 ;
  - pas de *perfect forward secrecy*.

- **Authentification du serveur**
  - initialement inexistante ;
  - puis clé publique signée par une clé privée **documentée** . . .
  - . . . donc inutile.

Mécanismes *Standard Security* (2/2)

- **Intégrité**
  - jusqu'à 5.1 inclus : simple MAC sur les données **en clair** ;
  - à partir de 5.2 : MAC sur les données en clair, avec un compteur.

- **Chiffrement**
  - RC4 :
    - 40, 56 ou 128 bits,
    - logique de choix de la taille complexe,
    - par défaut : taille choisie par le client ;
  - FIPS : triple DES.

- **Conclusion**
  - mécanismes de sécurité *propriétaires* ;
  - nécessité d'évolution.

Table des matières

Mécanismes *Enhanced Security*

TLS :

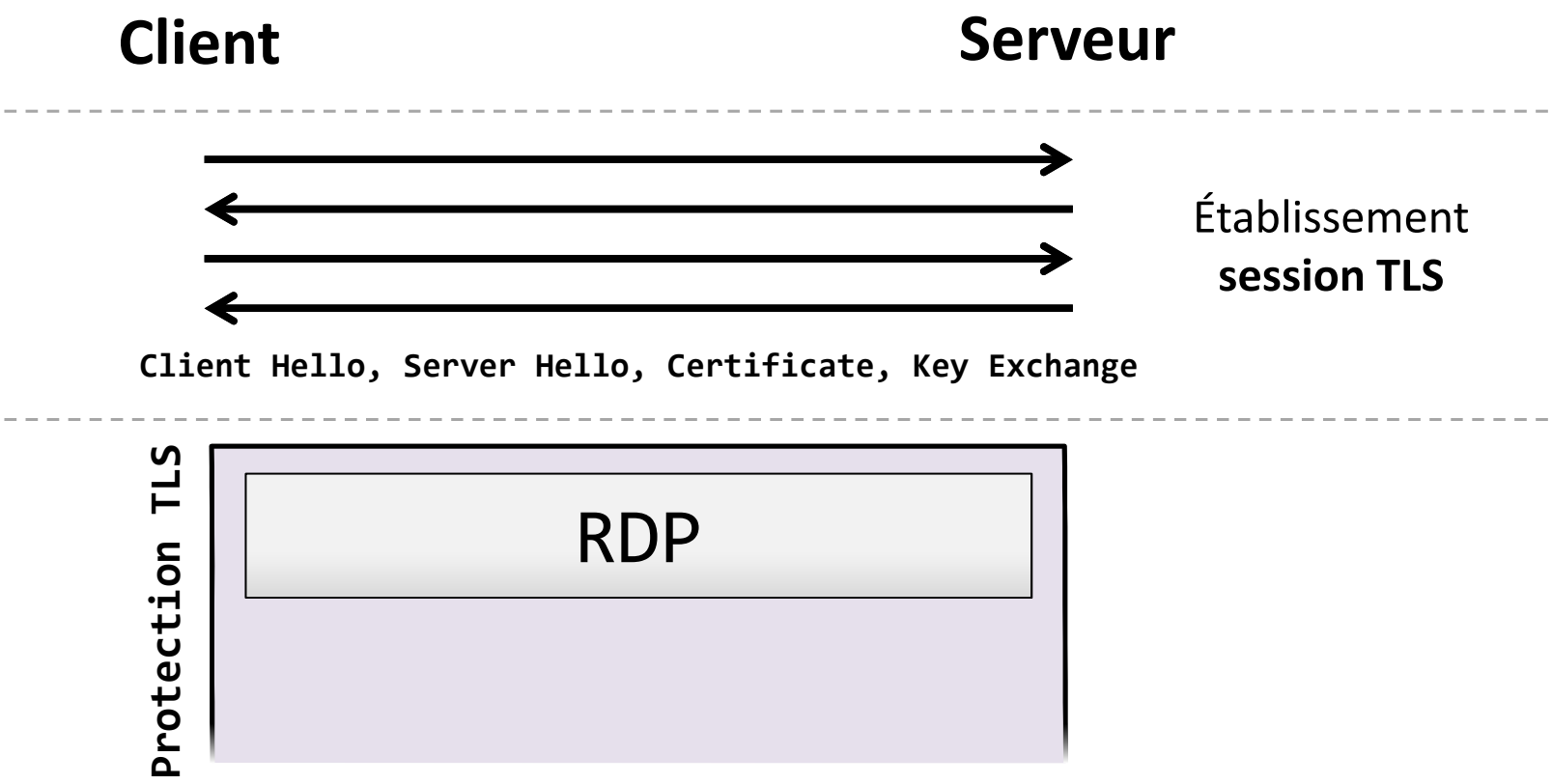
- introduit Windows 2003 SP1 ;
- permet l'authentification du serveur ;
- authentification TLS par certificat client non supportée.

NLA [1]/CredSSP :

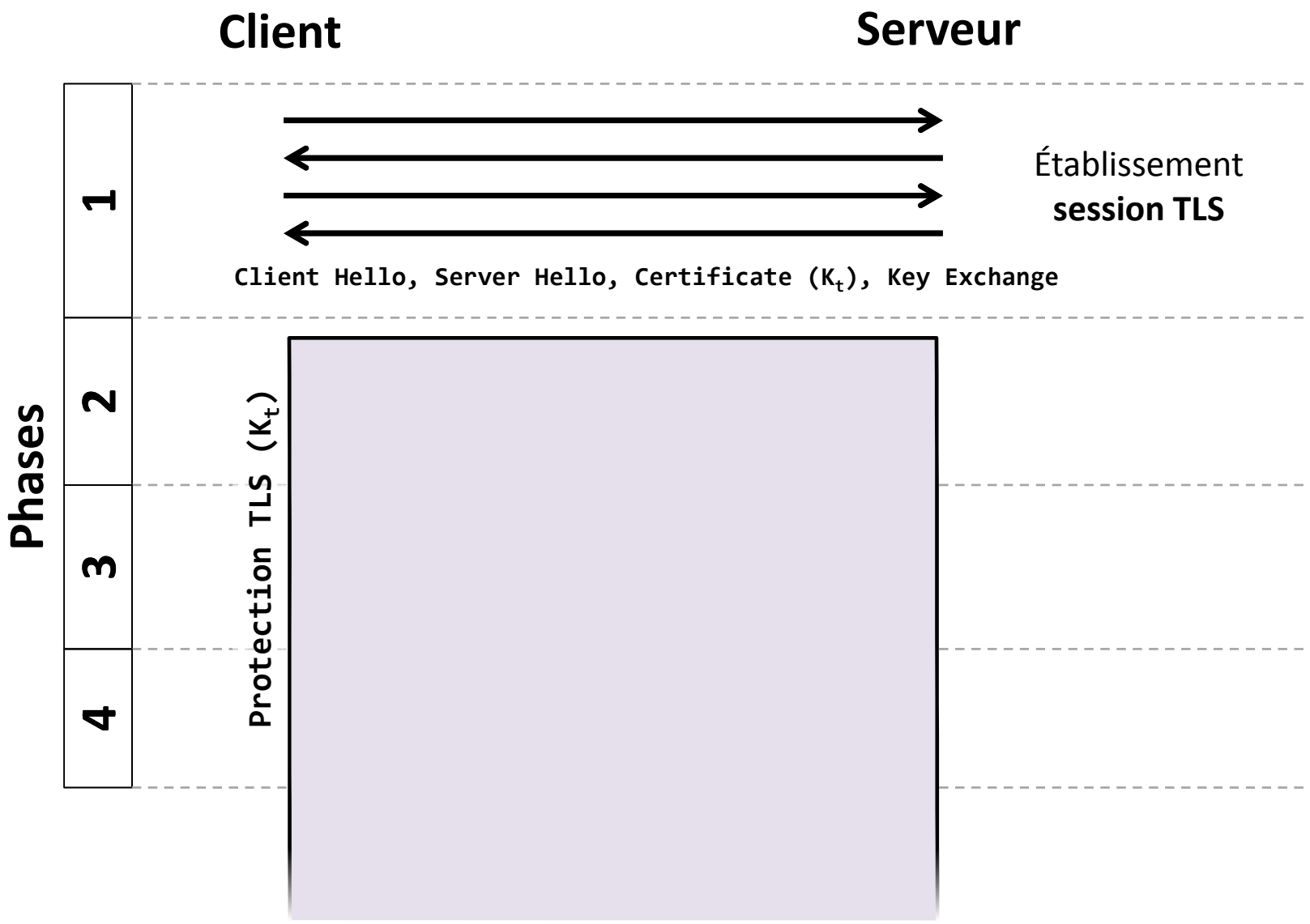
- introduit avec Windows Vista/2008 ;
- intègre l'authentification au protocole ;
- permet la délégation des authentifiants au serveur ;
- permet l'authentification du serveur par Kerberos.

1. *Network Level Authentication*

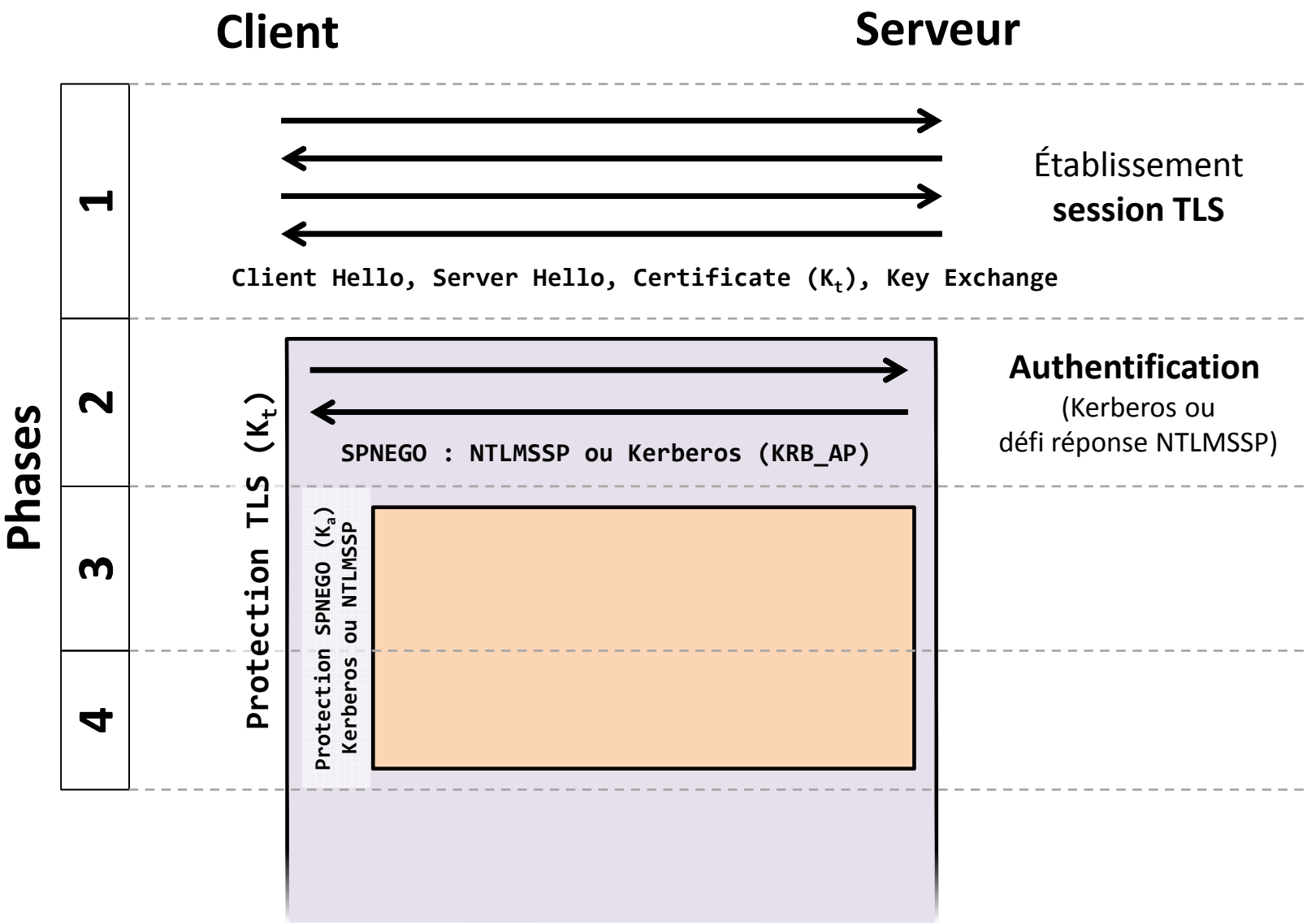
Couches protocolaires avec TLS



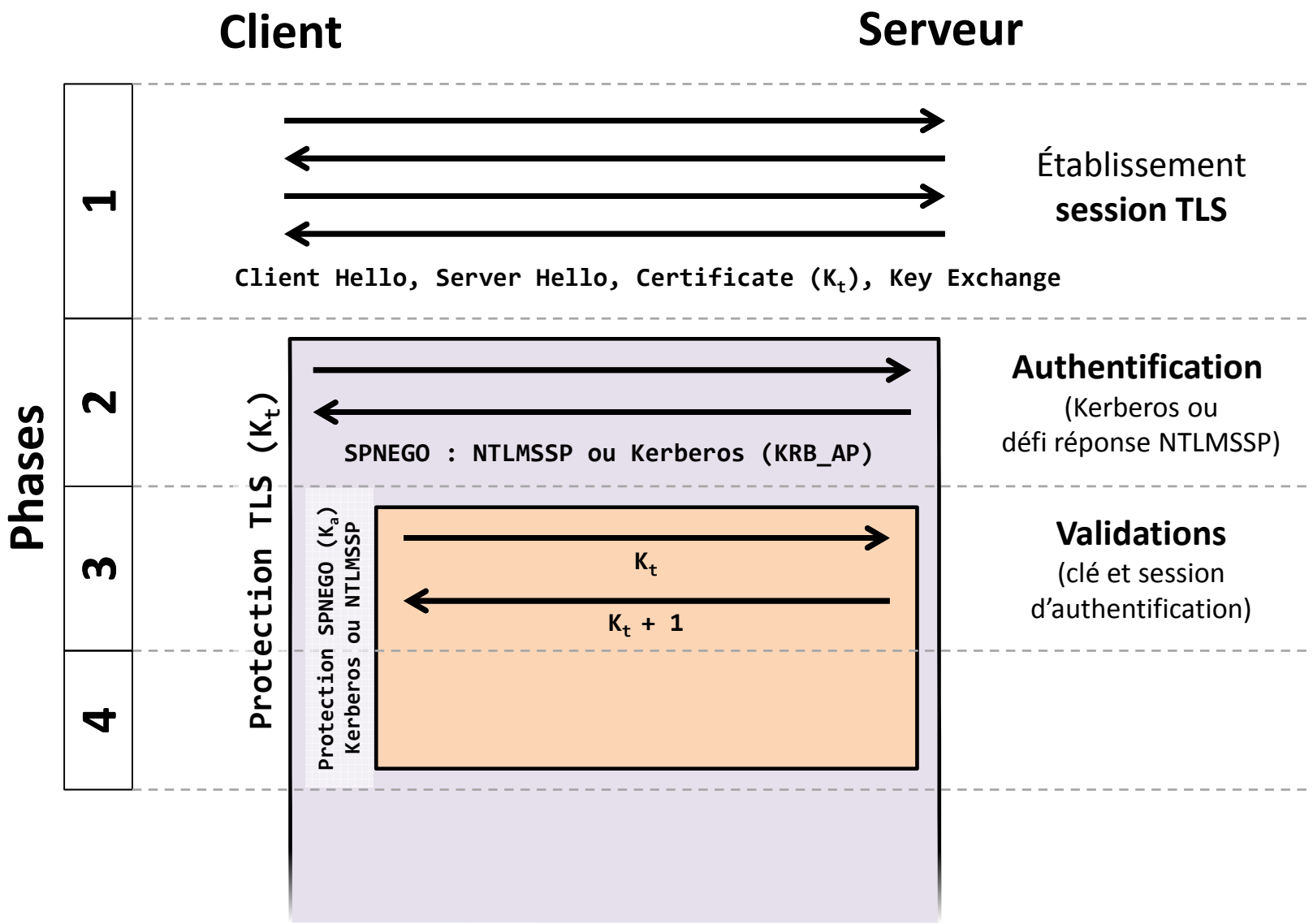
Couches protocolaires de NLA (1/4)



Couches protocolaires de NLA (2/4)



Couches protocolaires de NLA (3/4)



Couches protocolaires de NLA (4/4)

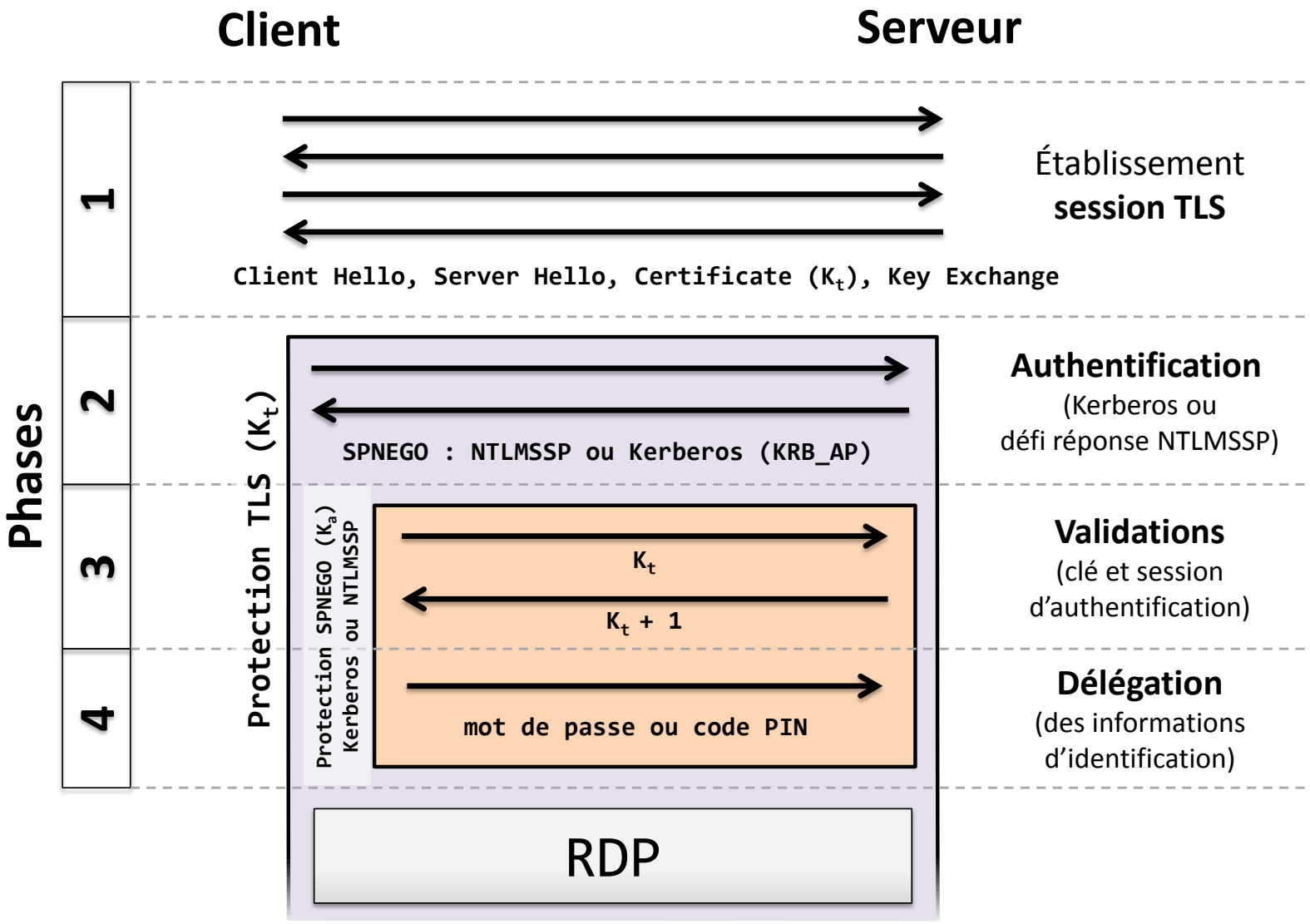
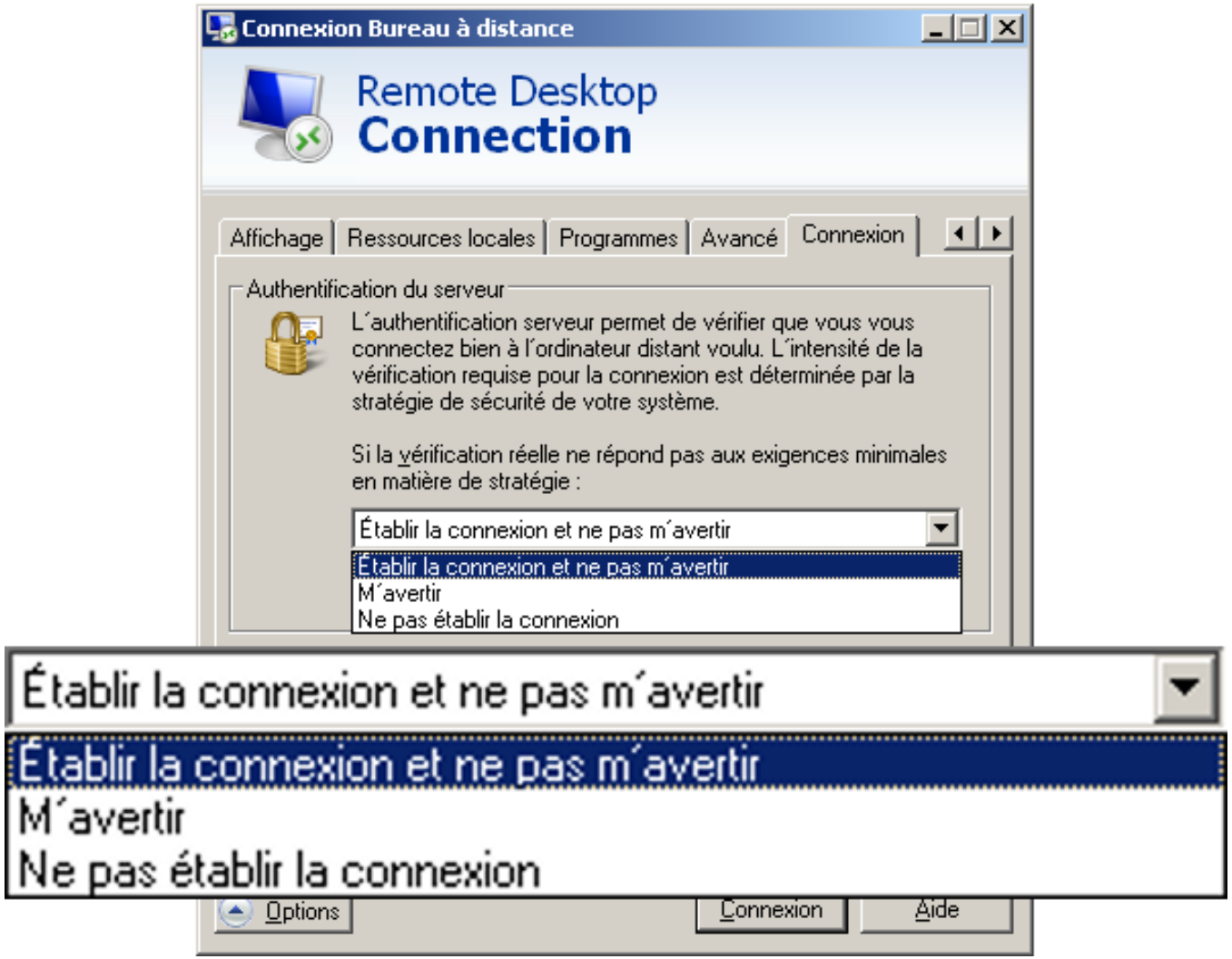
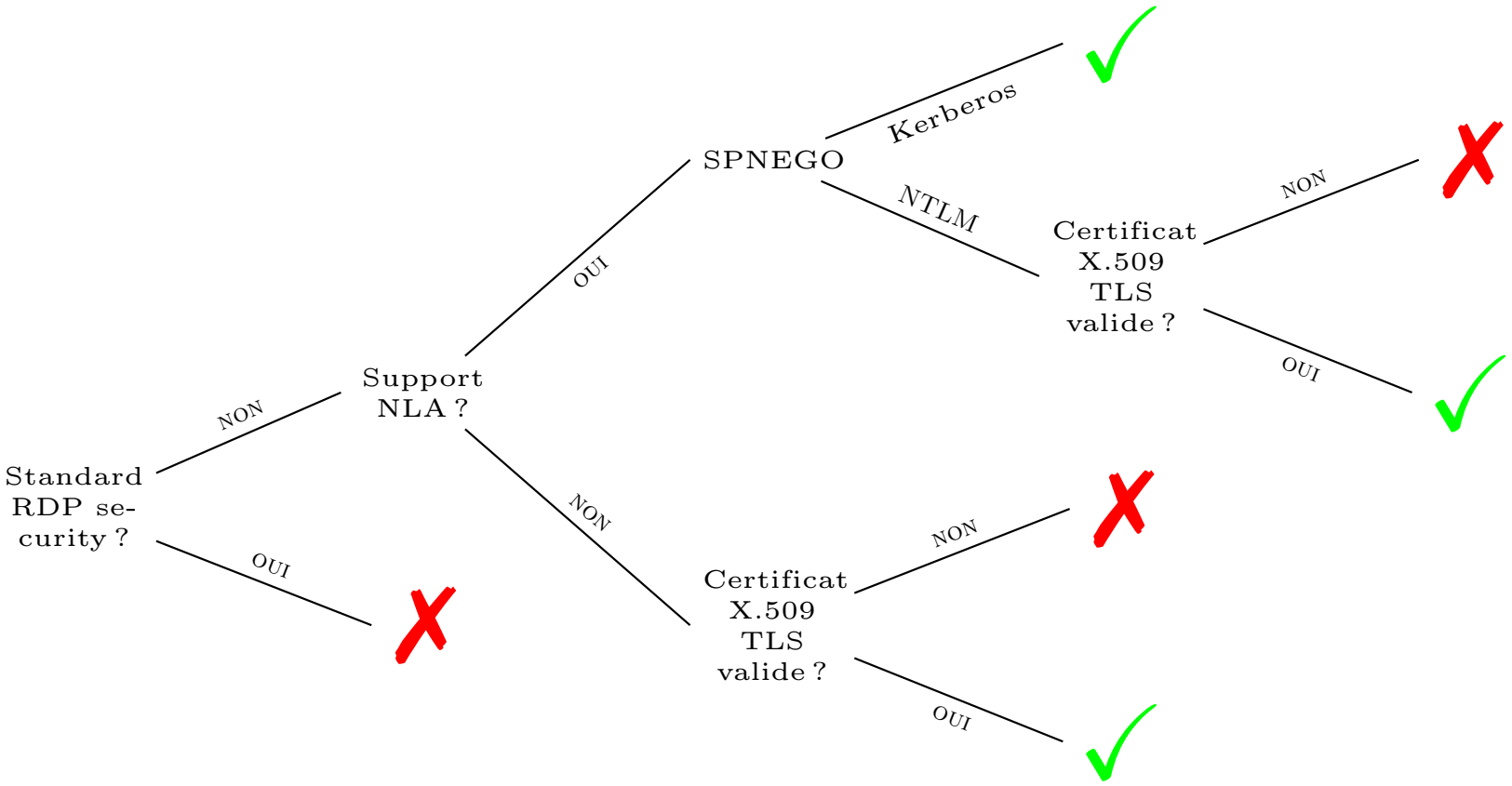


Table des matières

Options de configuration du client MSTSC



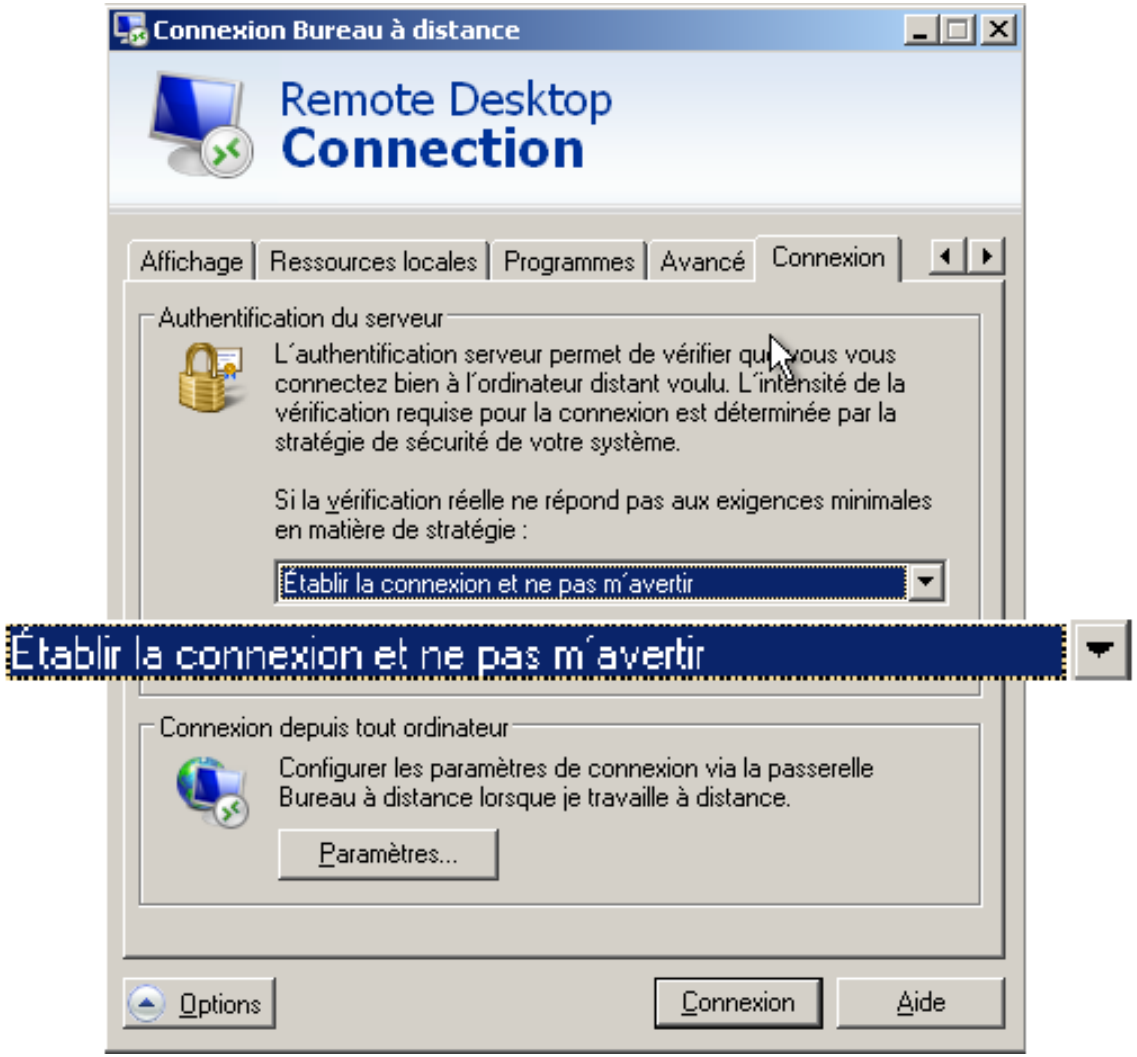
Logique d'authentification du serveur en *Enhanced*



- ✓ : Serveur authentifié
- ✗ : Serveur NON authentifié
- Note : le client 6.0 considérait une connexion NLA en NTLM comme authentifiée

Interlude récréatif #2

Configuration XP par défaut



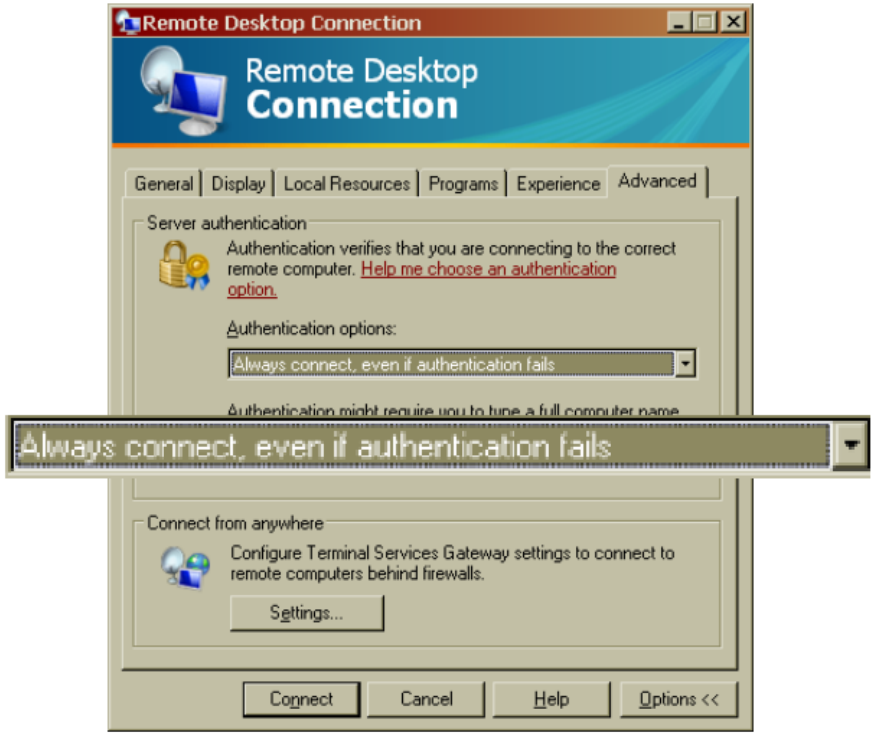
Interlude récréatif #3

Extrait du blog RDS [2]

How to eliminate the 'Remote Desktop cannot verify the identity of the computer you want to connect to..." messages

Answer: Before connecting, in Remote Desktop, do the following:

1. Click on "Options"
2. Click on the "Advanced Tab"
3. In "Authentication Options", select "Always connect, even if authentication fails, as seen below:



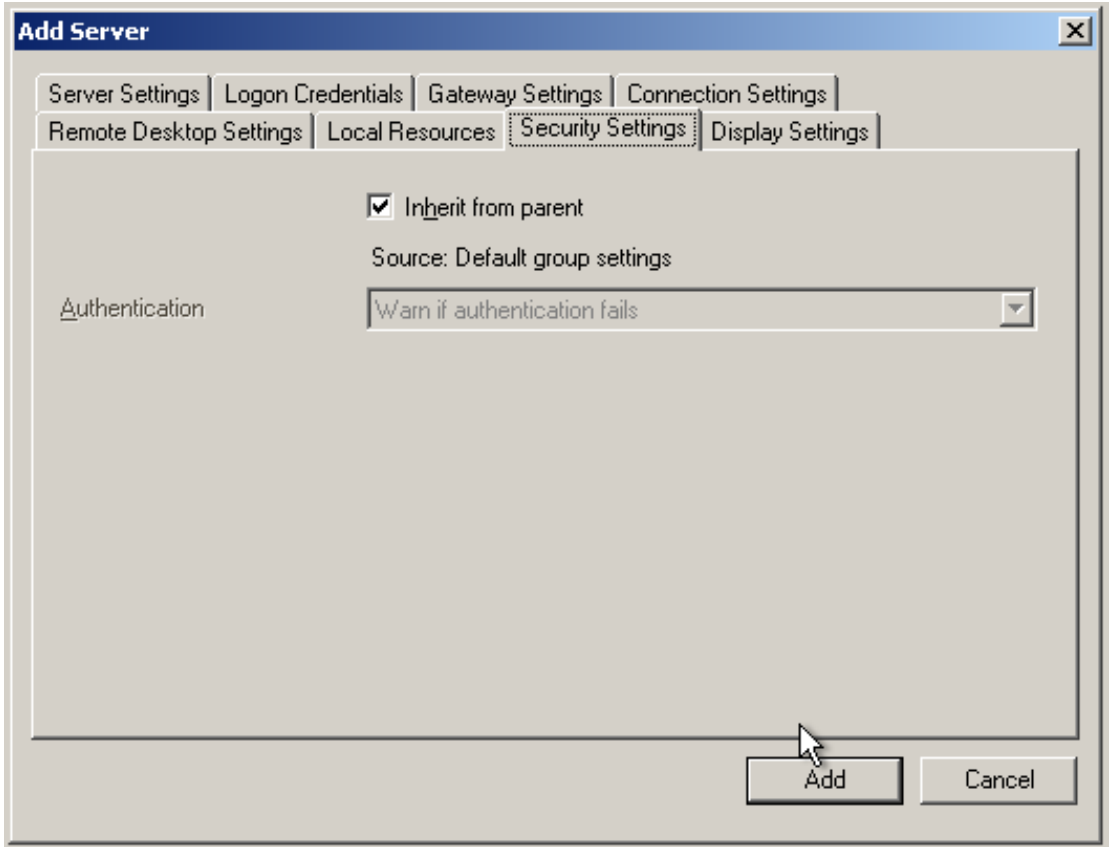
This will disable the warning prompt. Please be aware that selecting this option makes it possible for attackers to intercept and modify the data exchanged between client and server.

2. http://blogs.msdn.com/b/rds/archive/2007/01/22/vista-remote-desktop-connection-authentication-faq.aspx

Interlude récréatif #4

RDP connection manager

Vidéo présentant la logique d'héritage



Logique de connexion du client Microsoft

En NLA avec SPNEGO/NTLMSSP et l'option "M'avertir"

1. connexion TLS **sans validation du certificat du serveur**
2. échange SPNEGO/NTLMSSP dans la session TLS
3. échange $K_t/K_t + 1$
4. validation du certificat contre les ancres du système
   - 4.1 ✓ certificat validé : session RDP authentifiée
   - 4.2 ✗ certificat non validé : **déconnexion temporaire**

5. dans ce dernier cas, tentatives additionnelles de validation :
   - alerte et acception du certificat par l'utilisateur **ou**
   - validation par rapport aux empreintes de la base de registre

   en cas de succès :

6. **nouvelle** connexion TLS, **<u>sans aucune authentification</u>**
7. échange SPNEGO/NTLMSSP
8. délégation des identifiants
9. établissement de la session RDP

Interlude récréatif #5

- Contexte :
  - client Windows 7 à jour vers un Windows 2008 R2 à jour ;
  - sans Kerberos ;
  - MITM réseau ;
  - client configuré en “m’avertir” (**par défaut**).
- Pré-requis attaquant : rien.
- Problèmes sous-jacents :
  - logique de validation de l’authentification par le client MSTSC.

Vidéo

Recommandations (1/2)

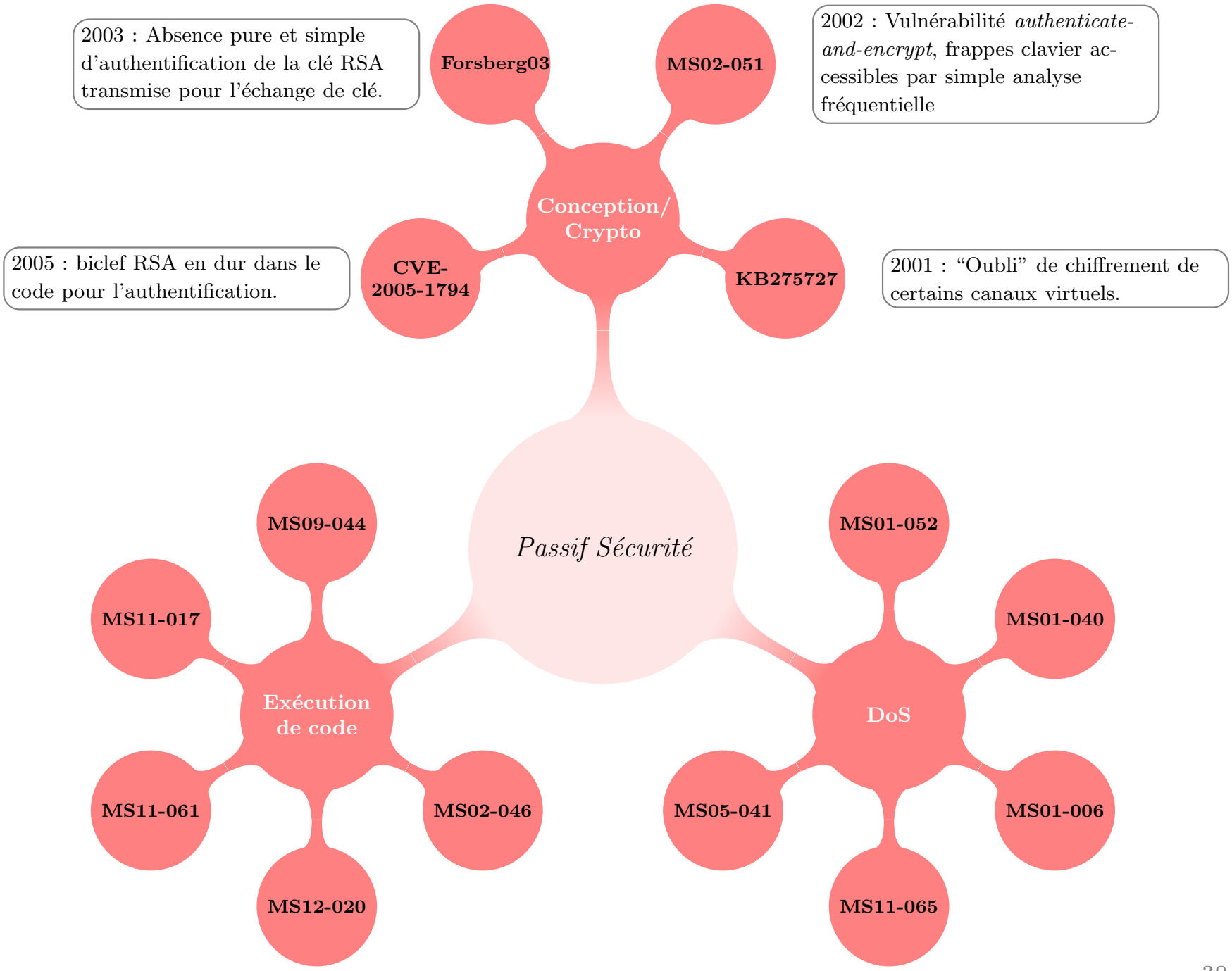
Configuration

Seules options possibles côté client :

- installer la dernière version du client Microsoft ;
- **forcer l'authentification du serveur** ("*Ne pas établir la connexion*").

Côté serveur :

- mise à jour impossible ;
- en domaine :
  - XP : pas de salut ;
  - sur 2003 : activer et forcer TLS ;
  - sur Vista, Seven, 2008 : forcer NLA.
- hors domaine :
  - XP : pas de salut ;
  - autres : forcer TLS.


2003 : Absence pure et simple d'authentification de la clé RSA transmise pour l'échange de clé.
Forsberg03
MS02-051
2002 : Vulnérabilité *authenticate-and-encrypt*, frappes clavier accessibles par simple analyse fréquentielle
Conception/ Crypto
2005 : biclef RSA en dur dans le code pour l'authentification.
CVE-2005-1794
KB275727
2001 : "Oubli" de chiffrement de certains canaux virtuels.
Passif Sécurité
MS09-044
MS11-017
Exécution de code
MS11-061
MS02-046
MS12-020
MS01-052
MS01-040
DoS
MS05-041
MS01-006
MS11-065

Recommandations (2/2)

Architecture

Réseau :

- service RDP accessible uniquement aux administrateurs ;
- protéger les flux RDP d'attaques réseau.

Conclusion :

- Nécessité d'un réseau d'administration dédié.

Questions ?