I'm listening to you !

Nicolas Bareil (nbareil@free.fr)

EADS CCR/SSI

2 juin 2005

Problématique

Beaucoup de projets VoIP

- Peu de gens connaissant les protocoles mis en jeux
- Faible conscience des nouvelles problématiques :
  - Numéro d'urgence
  - Besoin d'électricité EDF
  - etc.
- Notamment concernant la sécurité :
  - Intégrité
  - Disponibilité
  - Confidentialité

M. X, RSSI

« Vu le prix qu'on paie, on pense que c'est sécurisé »

Qu'en est-il concrètement ?

Problématique

Nous allons donc aborder :

- Un aperçu des protocoles mis en jeu
- À travers `ilty`, les attaques concrètes
- Les mesures limitant les dégats

Plan

Qu'est-ce que c'est ?

VoIP signifie *Voice over IP*, ou *téléphonie sur IP*

- Fonctionnalités identiques à la téléphonie classique
- Le support est le réseau IP existant
- Les principaux protocoles :
  - SIP, standardisé par l'IETF
  - Skinny, protocole propriétaire de Cisco
  - H.323

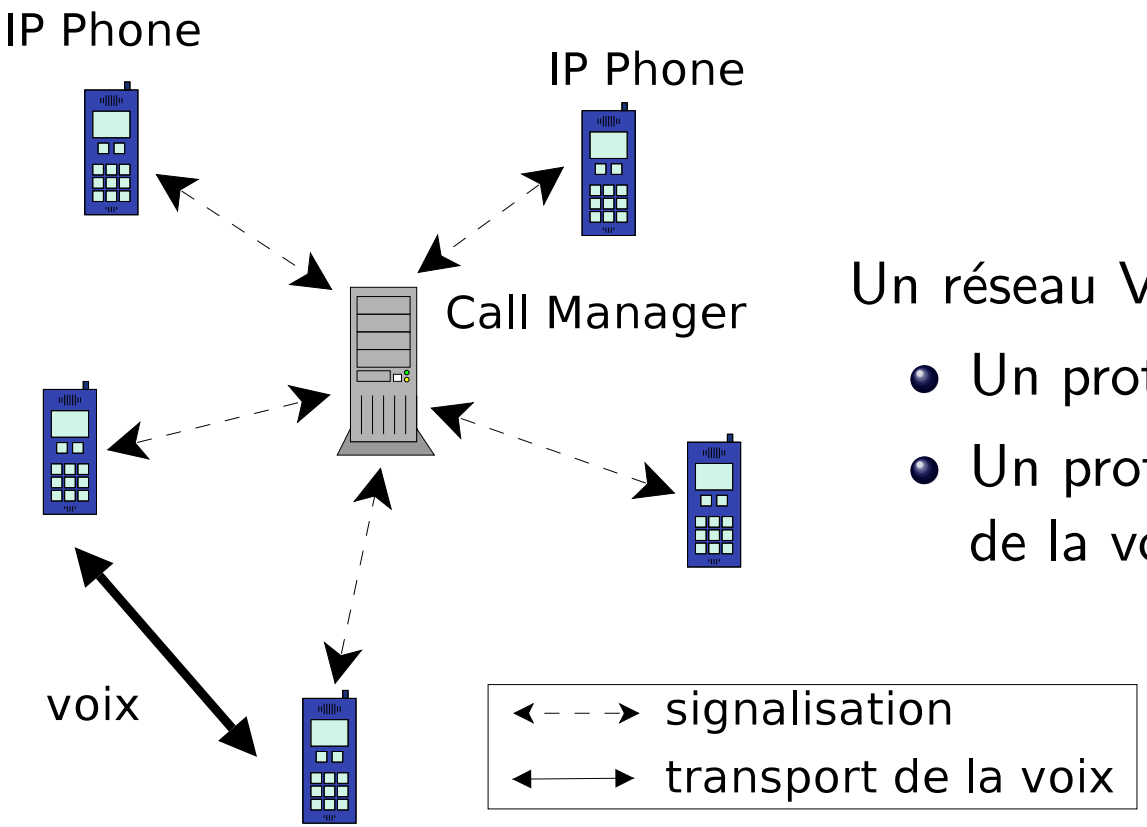
Équipements

Un réseau VoIP est constitué :

- des téléphones adaptés
- un entremetteur, le *Call Manager*

C'est également ce service qui fournit la passerelle vers le réseau classique (appel extérieur).

Les protocoles mis en jeux



Un réseau VoIP est géré via :

- Un protocole de signalisation
- Un protocole de transport de la voix

Présentation générale

Skinny :

- Utilise TCP/2000
- Protocole binaire (contrairement à SIP)
- Champs à positions fixes

```
0x00 : 0000 1234 cafe 300d beef 81da 8100 60bb
0x10 : 0800 4560 0040 33f5 0000 4006 6828 2a1b
0x20 : 4502 2a1b 4503 c93a 07d0 e65a e15d a9c7
0x30 : 439a 5018 0578 6de8 0000 1000 0000 0000
0x40 : 0000 0300 0000 0600 0000 0100 0000 bdf1
0x50 : 0002
```

Longueur du message totale : **16**

Présentation générale

Skinny :

- Utilise TCP/2000
- Protocole binaire (contrairement à SIP)
- Champs à positions fixes

```
0x00 : 0000 1234 cafe 300d beef 81da 8100 60bb
0x10 : 0800 4560 0040 33f5 0000 4006 6828 2a1b
0x20 : 4502 2a1b 4503 c93a 07d0 e65a e15d a9c7
0x30 : 439a 5018 0578 6de8 0000 1000 0000 0000
0x40 : 0000 0300 0000 0600 0000 0100 0000 bdf1
0x50 : 0002
```

Type de message : **AppuiSurUneTouche**

Présentation générale

Skinny :

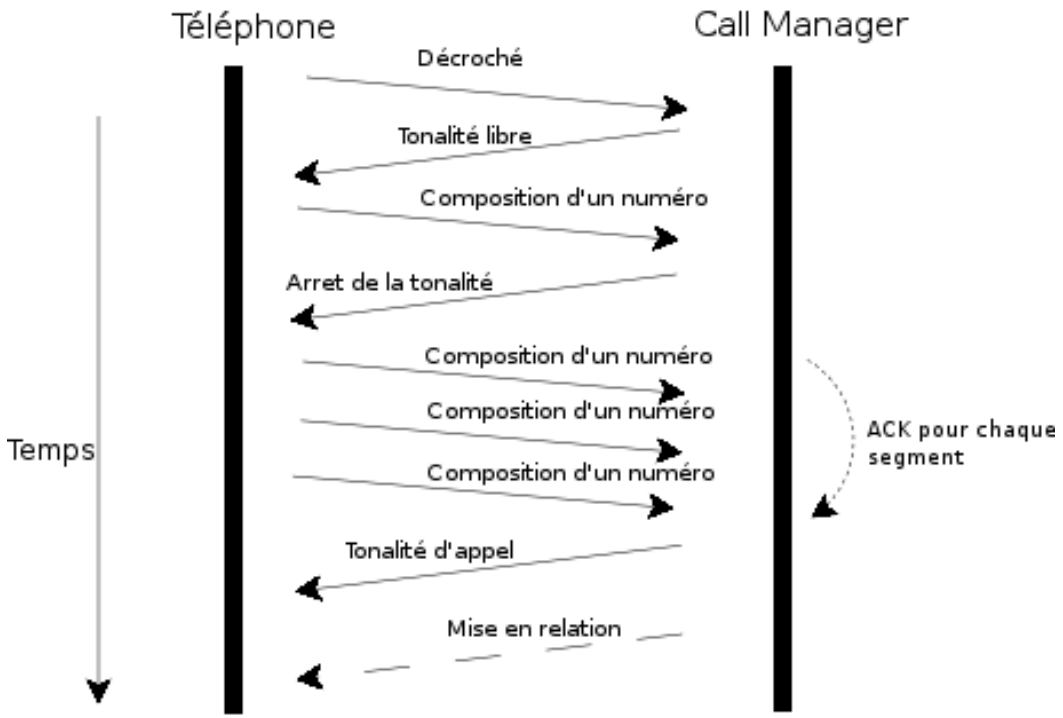
- Utilise TCP/2000
- Protocole binaire (contrairement à SIP)
- Champs à positions fixes

```
0x00 : 0000 1234 cafe 300d beef 81da 8100 60bb
0x10 : 0800 4560 0040 33f5 0000 4006 6828 2a1b
0x20 : 4502 2a1b 4503 c93a 07d0 e65a e15d a9c7
0x30 : 439a 5018 0578 6de8 0000 1000 0000 0000
0x40 : 0000 0300 0000 0600 0000 0100 0000 bdf1
0x50 : 0002
```

Touche **6** du téléphone pressée

Déroulement

Chronogramme d'un appel :



Real Time Protocol

Le RTP utilise l'UDP :

- Perte de paquets non critique
- Question de performances

Il n'y a pas de ports fixes, ils sont négociés et échangés par le protocole de signalisation.

Real Time Protocol : en-têtes

```
0x00 0000 1234 cafe 300d beef 81da 8100 a0bb
0x10 0800 45b8 00c8 2ff2 0000 4011 6b42 2a1b
0x20 4502 2a1b 4501 47e2 4ad6 00b4 0000 8000
0x30 85dc 00f3 56d0 0a81 7ac5 fefe fefe fefe
0x40 feff ffff 7f7f 7f7f ffff ff7f 7f7f 7f7f
0x50 7f7f 7f7e 7e7e 7e7e 7e7e 7f7f 7f7f 7f7f
...
0xd0 fefe feff ffff 7fff ffff
```

Algorithme de codage : **G711**

Real Time Protocol : en-têtes

```
0x00 0000 1234 cafe 300d beef 81da 8100 a0bb
0x10 0800 45b8 00c8 2ff2 0000 4011 6b42 2a1b
0x20 4502 2a1b 4501 47e2 4ad6 00b4 0000 8000
0x30 85dc 00f3 56d0 0a81 7ac5 fefe fefe fefe
0x40 feff ffff 7f7f 7f7f ffff ff7f 7f7f 7f7f
0x50 7f7f 7f7e 7e7e 7e7e 7e7e 7f7f 7f7f 7f7f
...
0xd0 fefe feff ffff 7fff ffff
```

Numéro de séquence : Détection des pertes

Real Time Protocol : en-têtes

```
0x00 0000 1234 cafe 300d beef 81da 8100 a0bb
0x10 0800 45b8 00c8 2ff2 0000 4011 6b42 2a1b
0x20 4502 2a1b 4501 47e2 4ad6 00b4 0000 8000
0x30 85dc 00f3 56d0 0a81 7ac5 fefe fefe fefe
0x40 feff ffff 7f7f 7f7f ffff ff7f 7f7f 7f7f
0x50 7f7f 7f7e 7e7e 7e7e 7e7e 7f7f 7f7f 7f7f
...
0xd0 fefe feff ffff 7fff ffff
```

Timestamp : Gestion du retard, de la gigue

Real Time Protocol : en-têtes

```
0x00 0000 1234 cafe 300d beef 81da 8100 a0bb
0x10 0800 45b8 00c8 2ff2 0000 4011 6b42 2a1b
0x20 4502 2a1b 4501 47e2 4ad6 00b4 0000 8000
0x30 85dc 00f3 56d0 0a81 7ac5 fefe fefe fefe
0x40 feff ffff 7f7f 7f7f ffff ff7f 7f7f 7f7f
0x50 7f7f 7f7e 7e7e 7e7e 7e7e 7f7f 7f7f 7f7f
...
0xd0 fefe feff ffff 7fff ffff
```

La voix

Real Time Protocol : voix

Après les en-têtes. . .la voix !

- La voix n'est pas chiffrée, juste codée
- Si on intercepte ces paquets, on peut donc écouter la conversation

À moins d'utiliser IPSec ou la version sécurisée de `RTP` (`SRTP`).

Plan

Savoir quoi détourner

- Le Call Manager en priorité
  - C'est la clef de tout le réseau
  - Adresse récupérable depuis une requête `DHCP` sur le réseau de téléphone
- Les téléphones
  - Récupérer les numéros par l'annuaire
  - Connaître les adresses grâce à la signalisation

Corruption de cache ARP

On se fait passer pour la cible en :

- Envoi de requêtes ARP falsifiées
- Répondant aux requêtes ARP en se faisant passer pour la cible
- Émettant de messages ARP *gratuitous*

Plutôt que de réinventer la roue, on utilise un programme tel que `arpspoof` (dsniff) ou `arp-sk`

Réponse aux requêtes DHCP

Le téléphone utilise le DHCP pour récupérer les informations de base :

- Son adressage réseau
- L'adresse du Call Manager

Répondons alors **avant le vrai serveur** en indiquant que nous sommes le Call Manager

Mise à jour de la configuration par TFTP

Au démarrage, le téléphone récupère sa configuration via TFTP

- Récupération d'un fichier XML
- Téléchargement des sons de base (tonalités, son de touche)

Il est alors possible de se faire passer pour un serveur TFTP

Détection des appels

Pour repérer le flux RTP, on peut utiliser des heuristiques telles que la détection par :

- Une suite de paquets de taille fixe
- Un échange constant entre deux ports « hauts » UDP
- etc.

ilty préfère utiliser directement le protocole de signalisation.

La signalisation

Surveiller le protocole de signalisation permet :

- Reconnaître précisemment les appels
  - Accès aux informations de l'annuaire
  - Numéro de téléphone
  - Extension du numéro de téléphone
- Voir les touches composées
  - Capture du code de messagerie vocale
  - Numéros de carte bleue

vomit

`vomit`[1] de *Niels Provos*

- Décodage de Skinny et du `RTP` (`G.711`)
- Capture d'une unique conversation à la fois
- Capable d'injecter un fichier `WAV`
- « Simple sniffer »

[1] `http://vomit.xtdnet.nl/`

voipong

`voipong`[2] de *Murat Balaban*

- Limité au `RTP` également
- Capture directe depuis le réseau
- Capable d'enregistrer plusieurs conversations en parallèle

[2] `http://www.enderunix.org/voipong/`

CAIN

`CAIN`[3] de *Massimiliano Montoro*

- Tourne sous Microsoft Windows
- Surveille le `RTP` et `SIP`
- Supporte de nombreux algorithmes de codage
- Freeware, closed source

[3] `http://www.oxid.it/cain.html`

Présentation

Le projet ilty (I'm listening to you !) est une centrale d'écoute téléphonique

- Écrit en Python
- Une interface « user-friendly »
- Développé pour EADS CCR/SSI
  - Utilisé lors des tests d'intrusion
  - Utiliser pour sensibiliser les administrateurs

Fonctionnalités

ilty peut :

- Écouter une conversation en direct
- Enregistrer et lire une discussion
- Logguer les appels
- Détourner le trafic de voix et de signalisation via la corruption de cache `ARP`

Son objectif est d'être opérationnel sans besoin de configuration.

Traitement de la voix

Il faut :

- Décoder la voix
- Mixer les paquets de voix arrivant au même instant

Plusieurs méthodes pour décoder la voix :

- Implémenter les algorithmes de décodage
- Utiliser un programme externe

Décodage de la voix

Afin de décoder la voix, ilty utilise intensivement les tubes (*pipes*) :

- Décodage du `G.711` par `sox`
- Mixage des voix par Esound (`esd`)

Exemple :

```
sox -Ub -r 8000 -t .raw - -t .ub - | esdcat -b -m -r 8000
```

Décodage de la voix

Afin de décoder la voix, ilty utilise intensivement les tubes (*pipes*) :

- Décodage du `G.711` par `sox`
- Mixage des voix par Esound (`esd`)

Exemple :

```
sox -Ub -r 8000 -t .raw - -t .ub - | esdcat -b -m -r 8000
```

Plus qu'un sniffer

ilty ne se limite(ra) pas qu'à sniffer :

- Corruption de cache `ARP`
- Réponse aux requêtes `DHCP` des téléphones
- (Détournement de route ?)
- Historique des conversations et évènements Skinny

Plan

Démonstration

Plan

VLAN

Un VLAN est un domaine de broadcast Ethernet logique qui permet une

- Bonne compartimentation des différents réseaux
  - entre les machines et les téléphones
  - entre les téléphones eux-même

Mais...

- Problème du branchement sauvage (+ émission de messages CDP)
- Saut de VLAN possible en cas de mauvaise configuration

Empêcher la corruption de cache ARP

Quelques moyens d'empêcher la corruption de cache ARP :

- Cache ARP statique
- Utilisation de switches de niveau 2 et 3
- L'utilisation de PVLAN
- Mise en œuvre de arp* (module pour le noyau Linux)

Cryptographie

La meilleure solution est l'utilisation de la cryptographie !

- Besoin d'authentification
- Contrôle d'intégrité
- Chiffrement

Solutions :

- IPSec : chiffrement de toutes les données IP
- SRTP : chiffrement de la voix uniquement
  - Le protocole de signalisation n'est pas chiffré

Mais l'ajout de la cryptographie introduit une latence.

Conclusion

Conclusion :

- Le chiffrement devrait être une fonctionnalité de base
- Axes de développement d'ilty
  - réécriture en cours
  - découverte autonome du réseau

Des questions ?

Des questions ?

- Nicolas Bareil (nbareil@free.fr)

Merci de votre attention !

Des questions ?

Des questions ?

- Nicolas Bareil (nbareil@free.fr)

Merci de votre attention !