

Squid - ICAP

Gestion de l'accès à Internet

Stéphane Davy – Alcôve
stephane.davy@alcove.fr

Sommaire

- Contexte
- Le protocole ICAP
 - historique
 - principe
 - les requêtes ICAP
- Implémentation dans Squid
 - mise en oeuvre
 - références

Contexte

- Internet = milieu hostile:
 - technique : virus, code malicieux,...
 - éthique : contenu « inapproprié », voire illicite
- Plusieurs acteurs pour gérer ces aspects:
 - antivirus
 - proxy
 - firewall
 - filtrage d'URL...

Communication entre les différentes briques?

- Se limiter au protocole de base (HTTP)
 - pas de soucis de compatibilité
 - limitation dans la transmission d'information
 - dispositifs en série
- Accord entre éditeurs (plugins)
 - architecture plus souple
 - transmission d'information plus aisée
 - dépendances fortes entre produits

Communication entre les différentes briques?

- Utiliser un protocole commun : **ICAP**
 - ↳ Internet Content Adaptation Protocol
- Objectifs : permettre de travailler simplement sur les requêtes et sur leur contenu
 - ↳ Remote procedure call on HTTP messages

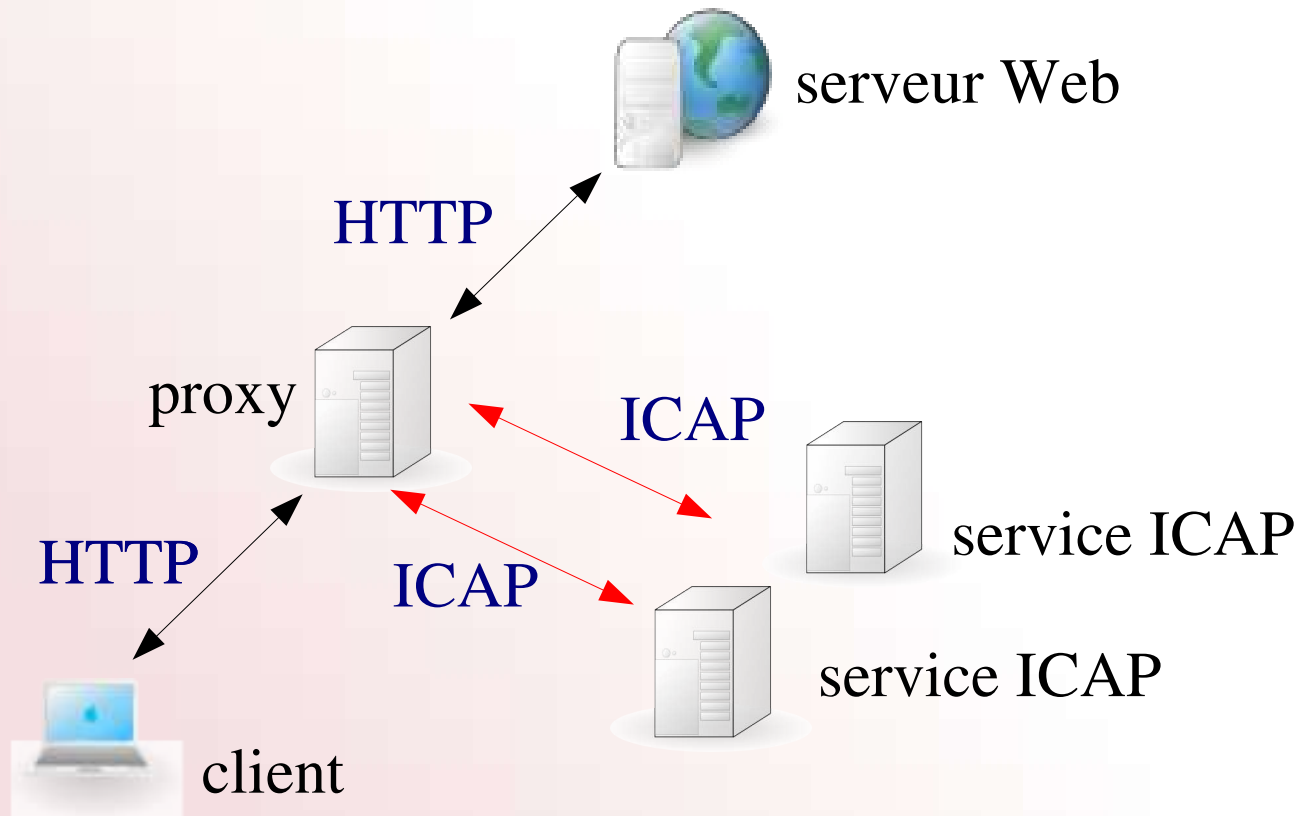
Le protocole ICAP

- Historique
- Principe
- Les requêtes ICAP
- Quelques exemples

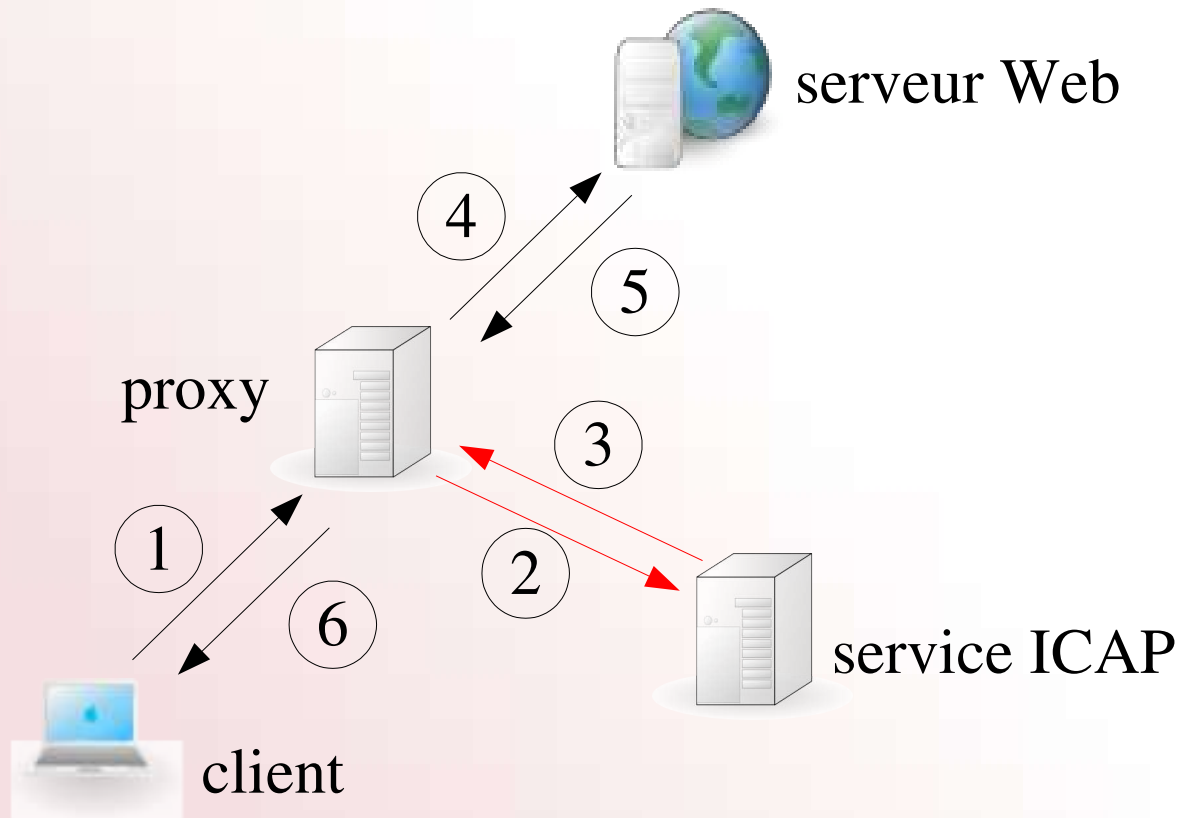
Le protocole ICAP : historique

- Initiative d'éditeurs : Akamai, Lucent Technologie, Network Appliance, Novell, WebWasher
- draft, puis rfc 3507 en Avril 2003
- Initiative parallèle à OPES
- ICAP/0.9: HTTP + entêtes spécifiques
- ICAP/1.0: dernière version

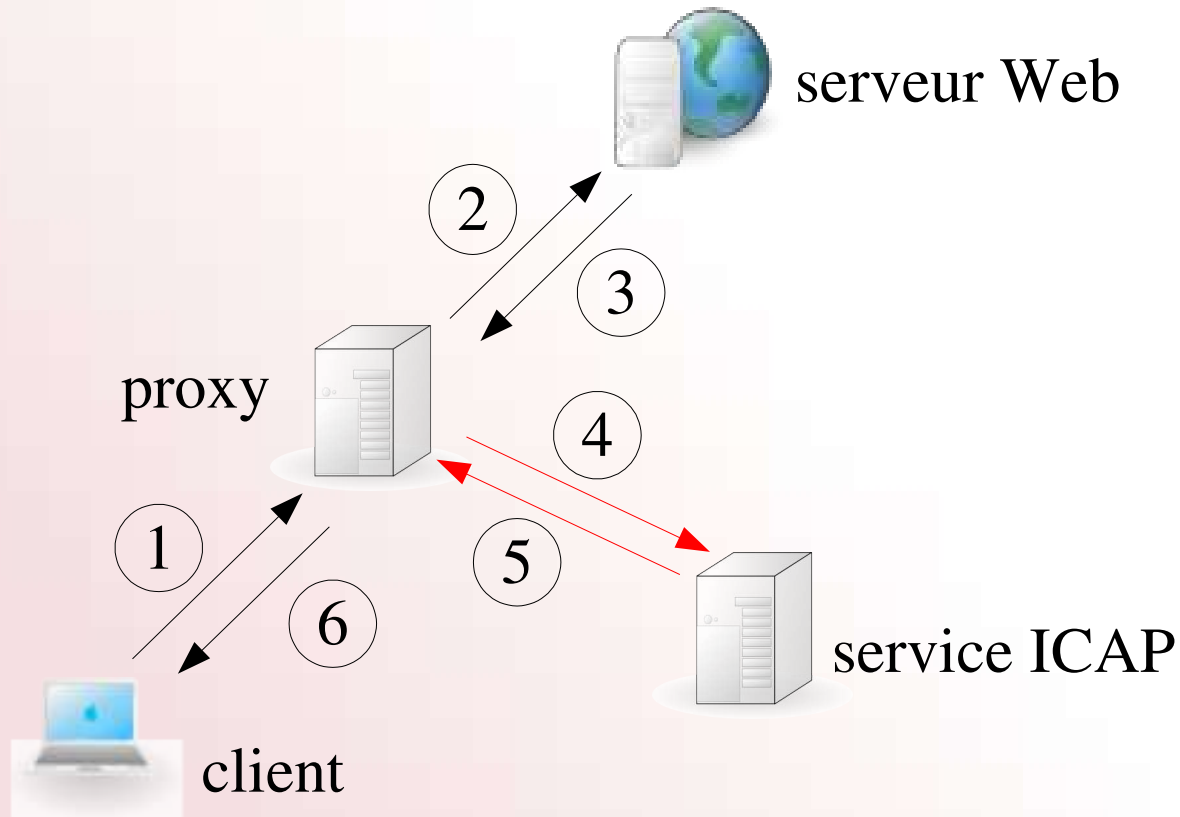
Le protocole ICAP : principe



Le protocole ICAP : « request mode »



Le protocole ICAP : « response mode »



Protocole ICAP : les requêtes

- Protocole requête / réponse ressemblant à HTTP
- utilisation d'URL : `icap://server:port/service`
- Envoi d'une requête, d'un entête et d'un corps :

```
REQMOD icap://icap-server:1344/service?arg=valeur ICAP/1.0
```

```
Host: icap-server
```

```
Encapsulated: req-hdr=0, null-body=170
```

```
GET / HTTP/1.0
```

```
Host: www.origin-server.com
```

Protocole ICAP : les requêtes (suite)

- le serveur retourne:
 - un code d'erreur
 - 100 : continuer après la «pr eview»
 - 204 : pas de modifications nécessaires
 - 40x : erreurs client
 - 50x : erreurs serveur
 - une requête modifiée ou
 - une réponse modifiée

Protocole ICAP : les requêtes (suite)

- les requêtes ICAP possibles :
 - REQMOD : modification de la requête HTTP
 - RESPMOD : modification de la réponse serveur
 - OPTIONS : liste des fonctionnalités du serveur
- « Preview » :
 - spécifié au client lors d'une requête OPTIONS
 - le client n'envoie qu'une partie du corps de la requête HTTP

Protocole ICAP : les requêtes (suite)

- transmission du message HTTP : nécessité d'indiquer la taille des entêtes et corps respectifs

=> champs encapsulated

Encapsulated: req-hdr=0, req-body=412

=> Requête en REQMOD, entêtes à offset 0, body à 412 octets (méthode POST)

Encapsulated: req-hdr=0, null-body=412

=> Requête en REQMOD, entêtes à 0, pas de body (méthode GET)

Protocole ICAP : gestion de l'authentification

- Information contenue dans les entêtes HTTP
- pas toujours exploitable
 - => authentification effectuée par le proxy
 - => transmission de l'information au serveur ICAP
- Mais: pas de normalisation pour l'instant:
champs **X-Authenticated-user**

Implémentation dans Squid

- historique
- exemples
- mise en oeuvre

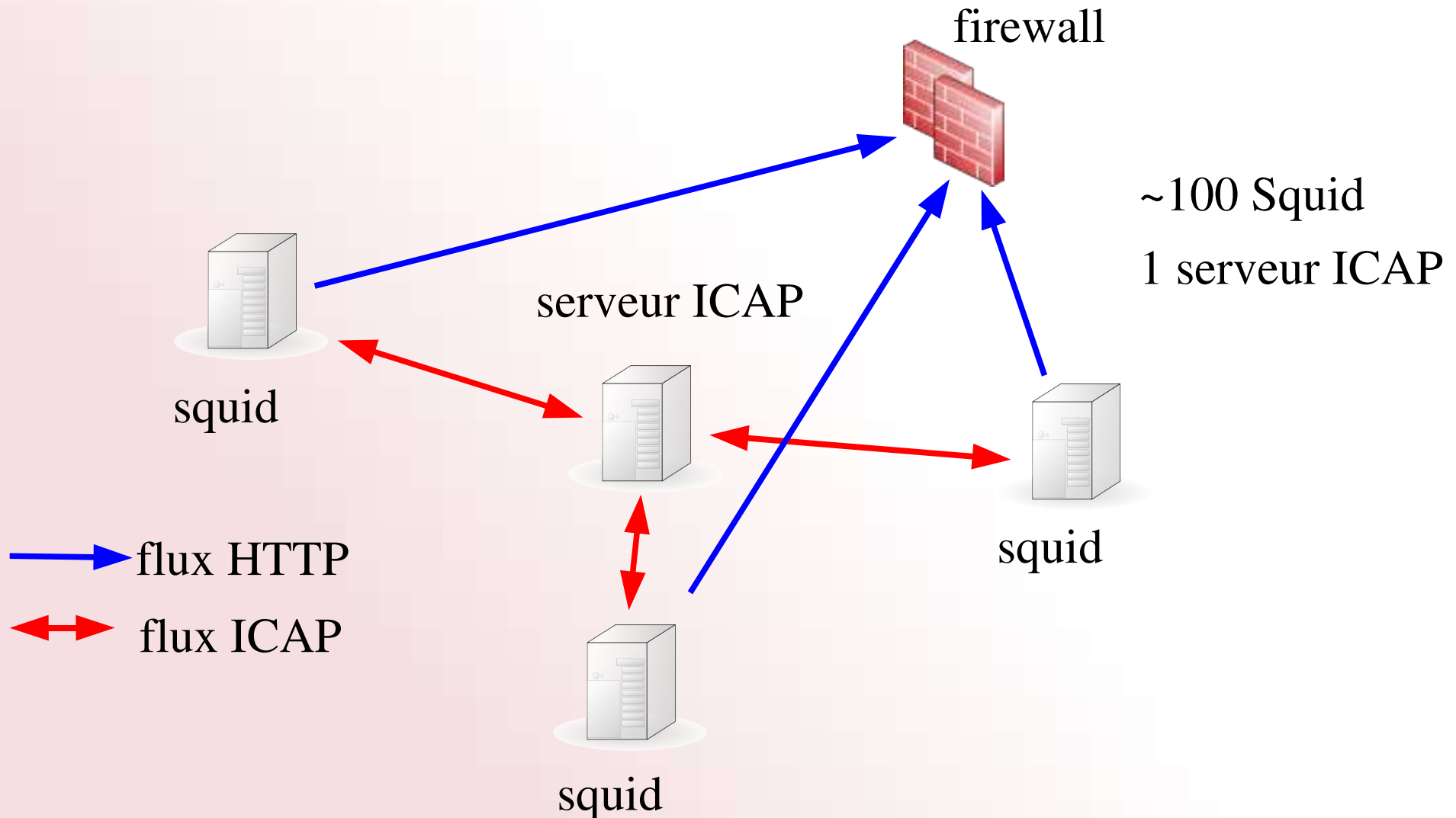
Squid et ICAP: historique

- premier « client » initié par HP
 - => code beta non utilisable
- contribution financière et technique de WebWasher
 - => première version stabilisée en REQMOD fin 2002
 - => RESPMOD fin 2003
- aujourd'hui en production

Squid et ICAP: exemples

- BICS : 2000 utilisateurs en REQMOD
- EDF : 30000 utilisateurs en REQMOD
- CEGEDIM: 2500 utilisateurs en REQMOD+RESPMOD
- ALTIS : 1500 utilisateurs en REQMOD+RESPMOD
- CNAM

Squid et ICAP: exemples



Squid et ICAP : fonctionnalités

- REQMOD
- RESPMOD
- « bypass »
- répartition de charge
- haute disponibilité
- transmission du X-Authenticated-user
- preview: code 204 non géré correctement

Squid et ICAP: mise en oeuvre

- Plusieurs sources disponibles
- mais une seule « fiable »
 - => <http://devel.squid-cache.org/projects.html#icap>
- mise en place peu simple:
 - automake version 1.5
 - autoconf version 2.13

Squid et ICAP: mise en oeuvre

- récupérer les sources Squid et le patch
- appliquer le patch
- lancer bootstrap.sh
- `configure --enable-icap-support`

Squid et ICAP: mise en oeuvre

Directives dans squid.conf :

```
icap_enable on | off
```

```
icap_send_auth_user on | off
```

```
icap_service service_1 reqmod_precache 1 icap://icap1:1344/reqmod
```

```
icap_service service_1 reqmod_precache 1 icap://icap2:1344/reqmod
```

```
icap_service service_2 respmod_precache 0 icap://icap3:1344/respmod
```

```
icap_class class_1 service_1 service_2
```

```
icap_access class_1 allow all
```

Conclusion

- ICAP
 - protocole ouvert
 - introduit de la souplesse
 - adopté par plusieurs éditeurs
 - évolutions à venir
 - extension à SMTP

Conclusion

- Squid / ICAP
 - stabilisé
 - utilisable en production
 - manque quelques fonctionnalités
 - quelques problèmes subsistent
 - manque de ressource sur le projet
 - manque de serveur ICAP libre?

Ressources

- RFC 3507
- www.i-cap.org: forum ICAP
- **SOURCE:** <http://devel.squid-cache.org/projects.html#icap>
- **mailing list:** <https://lists.sourceforge.net/lists/listinfo/icap-server-squidclient>