Captive-portal e Laboratori didattici: l'esperienza di UNICH

Dr. Damiano Verzulli

APM – Università “G' d'Annunzio”
Chieti - Pescara

damiano@verzulli.it

In collaborazione con:
CINECA
Consorzio Interuniversitario

Il problema

Da un certo punto di vista, alcuni (quanti?) segmenti di una Rete d'Ateneo possono essere visti come “porti di mare”:

- persone (*studenti?*) che accedono a macchine “wired” con accesso **illimitato** ad Internet, senza alcuna politica;
- persone che usano notebook (*e netbook; e smartphone; etc.*) per collegarsi a reti **wireless**... trovate per caso;
- docenti che, alla fine dell'anno, attivano un'**aula didattica** da 20 posti... in totale autonomia;
- altre situazioni analoghe alle precedenti...

Chiaramente questo scenario è riconducibile a:

- docenti/ricercatori/dottorandi/personale (ma, nel nostro caso, anche studenti) “smanettoni”, che spendono 30 dei propri euro per comprare e collegare AP o router wireless (o qualche euro in più per qualche PC o eventuali interi nuovi laboratori didattici)

L'APM è il primo soggetto chiamato a rispondere per ogni eventuale abuso (PolPost et al.)

La soluzione è “organizzativa” e tipicamente non nel controllo dell'APM (almeno a Chieti)

Approccio

Per risolvere (parzialmente) i problemi citati, abbiamo deciso di procedere per passi:

STEP 1 - MONITORING: introduzione di una serie di sistemi volti a “segmentare” (VLAN), “gestire” (firewalling) e “monitorare” (flussi) il traffico in transito nei vari segmenti della Rete d'Ateneo, con successiva analisi (umana) dei relativi Report

Così facendo abbiamo ottenuto:

- una drastica riduzione del traffico “critico” (P2P & al.);
- la possibilità di associare **facilmente** ogni flusso di traffico alla coppia IP/MAC_ADDRESS di origine, **anche** in presenza di NAT;
- la “sensazione” di controllo sugli abusi, soprattutto in presenza di volumi elevati.

È un pregio... ma anche un limite, nei contesti dove la postazione (IP/MAC) non è riconducibile ad un utente specifico**:**
Laboratori, WLAN, Postazioni Pubbliche, etc.

Approccio

...e quindi?

STEP 2 – AAA per le subnet critiche: introduzione di una politica di autenticazione degli utenti e relativo logging delle informazioni necessarie ad associare l'utente al traffico generato

La misura è “invasiva” in quanto “rompe” uno standard presente da sempre: l'utente fa tutto, senza controllo.

Per questo motivo, si è partiti dai segmenti più critici (*studenti – tra l'altro meno politicamente problematici*).

Il primo test-bed (ormai in corso da qualche anno) ha interessato:

- Nuvole Wireless “ufficiali”
- Laboratori Didattici

Sappiamo tutti che è anche un “requirement” normativo

(...che in pochi [nessuno?] sanno come mettere in pratica e di cui si parla pochissimo)

Soluzioni possibili...

Le macro-soluzioni che abbiamo analizzato sono state:

- Soluzioni complete/chiavi_in_mano di **NAC** (Network Access Controller);
- L'impiego di **802.1x**;
- Altre soluzioni più “light”

Non applicabile perchè:

- sebbene la quasi totalità delle porte “access” sia gestita da switch 802.1x compliant, non è possibile “sradicare” il proliferarsi di apparati incompatibili;
- difficoltà di gestione (Auth) della parte “wired”

Non applicabile perchè:

- troppo legate all'impiego di apparati di switching specifici (“vendor-lock-in”);
- cercavamo una soluzione scalabile a costo marginale quasi zero

Strada da seguire:

- Autenticazione effettuata via browser (**Captive-Portal**);
- Autorizzazione gestita server-side dal Captive Portal;
- Accounting “classico” ottenuto con **firewall** (solo porte proxabili) e uso forzato di **Proxy** (Squid - Transparent)

Captive Portal: quale?

Le soluzioni applicative testate (ed in produzione) sono:

www.nocat.net

NOCAT: installato nel 2006 per disciplinare l'accesso ad una piccola rete WiFi (18 AP) e, da allora, ancora in produzione;

www.chillispot.info

CHILLISPOT: installato nel 2008 per disciplinare l'accesso ad un laboratorio didattico. Successivamente esteso ad una WLAN ed a due ulteriori laboratori. Tuttora in produzione

Dopo un po' di ragionamento abbiamo capito che i Captive-Portal sono stati pensati per il mondo Wireless ma sono perfettamente impiegabili nel mondo WIRED!

analizziamo qualche dettaglio...

NOCAT *(...in breve)*

- Scritto in **PERL**, consiste di due componenti: **NocatAuth** e **NocatGateway**. L'idea è che ci sia un Auth-Server in grado di gestire più Gateway;
- Il Gateway sfrutta pesantemente **netfilter** e, in particolare, il **REDIRECT** (per “catturare” i client anonimi) ed il **MARK** (client autenticati vengono MARKati in modo diverso dai client anonimi);
- In funzione del MARK (oltre che dei source e dest IP/MAC/Porta e Prot), si può “giocare” con i pacchetti a proprio piacimento (WhiteList, BlackList, Profilazione, etc.). Ovviamente “giocare” direttamente con netfilter (= con iptables) non è banale;
- la soluzione “di default”, prevede soltanto **due (sole) tipologie di utenza:** Member (autenticati) e Public (non autenticati);
- Dopo una settimana di reverse-engineering+riscrittura, abbiamo “riorganizzato” il setup iniziale (iptables) e lo script di login per gestire anche il gruppo **staff**... a cui dare il **full-NAT :-)**
- Per massimizzare il “controllo”, abbiamo abilitato in uscita i soli protocolli **HTTP** e **FTP**, forzando gli stessi a passare per **SQUID** in modalità **trasparente**

NOCAT *(...in breve)*

VANTAGGI

Soluzione Open-Source, e quindi "adattabile" a piacere. Infatti abbiamo:

- "esteso" il sistema di profilazione (gruppo "staff");
- "esteso" il meccanismo di autenticazione (Auth via a-la-proxy-HTTP);
- Scalabile a costo marginale quasi nullo

SVANTAGGI

- La profilazione di base è insufficiente (member/public);
- Senza una padronanza "importante" di netfilter/iptables, è impensabile metterci sopra le mani;
- Mancanza di un supporto nativo alle VLAN (un GW su un HOST con N VLAN)

Ma ancora più importante:

- Progetto **ABBANDONATO** da tempo

Ad oggi, utilizzarlo è un ottimo esercizio didattico a meno di (ri)prenderlo in mano sul serio

ChilliSpot *(...in breve)*

- Scritto in **C**, è basato su un demone (**chilli**) che gestisce l'intero processo di gestione del traffico dei client;
- I client, di fatto, NON interagiscono con l'interfaccia fisica dell'host, ma bensì con una interfaccia virtuale (**tun***) gestita sempre dal processo chilli;
- l'interazione con il client, anche in fase di assegnazione di IP è gestita sempre da chilli (**DHCP "interno" e "legacy"**);
- Si appoggia a:
  - ✔ iptables per la gestione del traffico;
  - ✔ un proprio componente per la gestione di una parte delle attività web e del login;
- Supporta vari backend di AAA, fra i quali **Radius** (FreeRadius);
- Supporta il **traffic-shaping**;
- Per massimizzare il "controllo", abbiamo abilitato in uscita i soli protocolli **HTTP** e **FTP**, forzando gli stessi a passare per **SQUID** in modalità **trasparente**

ChilliSpot *(...in breve)*

VANTAGGI

Soluzione Open-Source, e quindi “adattabile” a piacere. Infatti abbiamo:

- adattato l'oggetto per gestire N istanze su un unico host (una istanza per ogni interfaccia fisica/VLAN)
- ottima integrazione con Radius;
- ottima gestione del traffic-shaping e dell'accounting;
- Scalabile a costo marginale quasi nullo

SVANTAGGI

- DHCP “legacy”;
- l'utilizzo di interfacce virtuali (tun*) complica abbastanza la gestione del networking;
- Difficoltà d'impiego in ambienti multi VLAN (occorre una istanza separata per ogni interfaccia)

Ma ancora più importante:

- Progetto **ABBANDONATO** da tempo
  (*c'e' il successore: CoovaChilli, che però non abbiamo investigato a fondo*)

Nonostante sia abbandonato, può essere impiegato utilmente, specie nei contesti più “piccoli”

Come procedere?

Alla luce delle fatiche e della insoddisfazione delle due soluzioni (difficoltà di scalare), abbiamo cercato soluzioni alternative.

Dopo ricerche piuttosto approndite, siamo arrivati a BCROUTER.

Presentato alla TNC2007, **BCRouter è una soluzione ideata dalla K.U.Leuven University** (Belgio), per la gestione del proprio network (KotNet).

www.bcrouter.net

Visti i numeri di KotNet (***30K studenti connessi; 500 Mbps di traffico supportati; 140 VLAN gestite*** *[numeri del 2005]*) la soluzione ha subito destato il nostro interesse...

BCROUTER *(...in breve)*

- Ideato per gestire ESATTAMENTE il nostro stesso problema *--in un contesto ben più complesso del nostro--* BCrouter nasce:
  - ✔ per impedire l'accesso anonimo ad Internet (**Network authentication**);
  - ✔ per limitare il numero di Gigabyte/mese per singolo utente (**Network quota enforcement**);
  - ✔ per evitare che pochi utenti consumino tutta la banda disponibile (**Network bandwidth regulation**);
- Non richiede software particolari client-side. Tutto viene gestito via web (Captive-Portal/HTTPS);
- Gestione contemporanea sia a livello di IP che di UTENTE;
- **Controllo della banda "real-time"**;
- **Bandwidth throttling**, con garanzia di non bloccare mai nessuno;
- Sistema di gestione delle eccezioni particolarmente sofisticato.

Troppo bello per essere vero?

BCROUTER *(...in breve)*

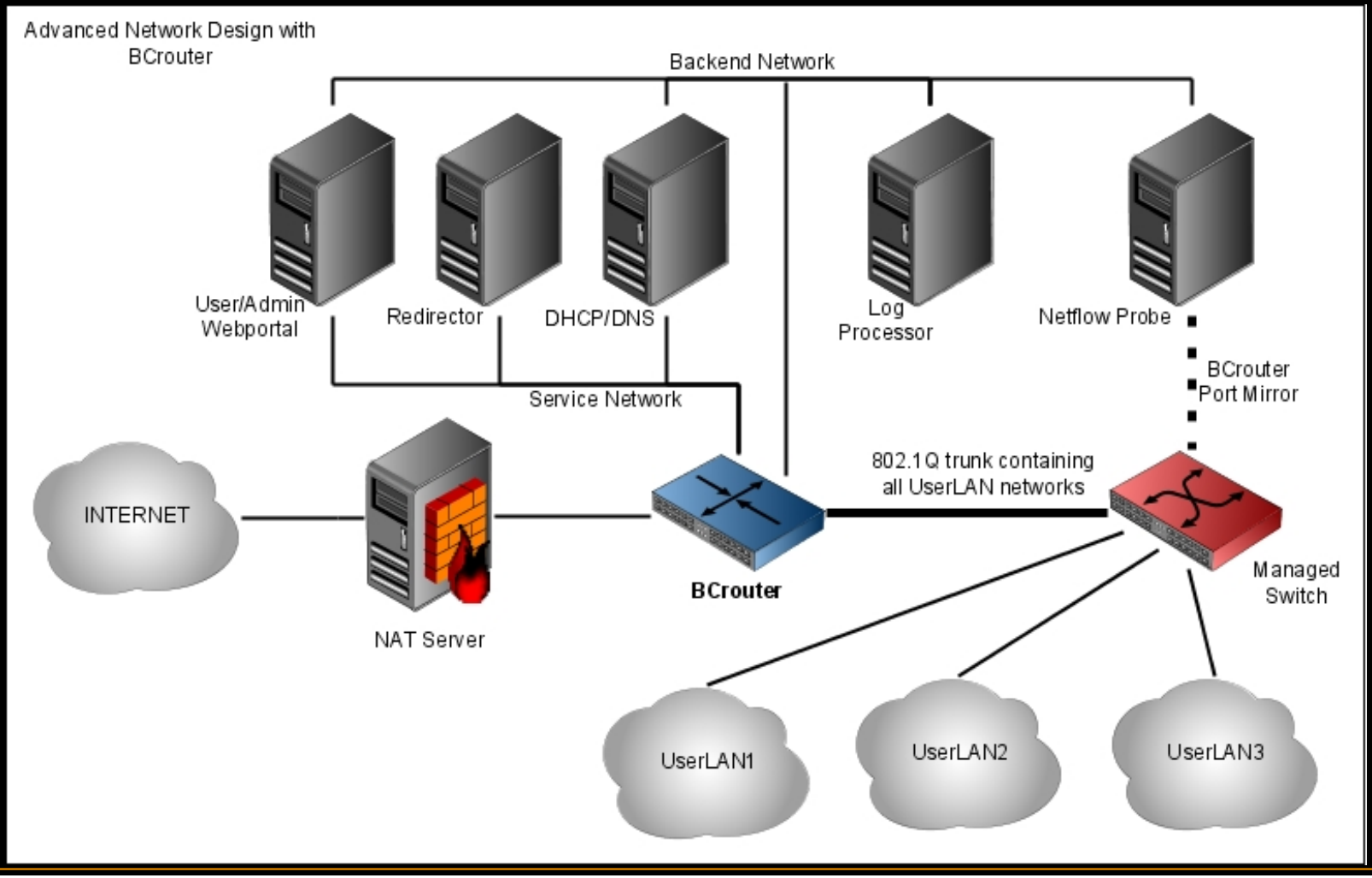
BCRouter riesce a fare quello che fa, grazie ad una architettura particolarmente innovativa:

- La gestione del traffico (accounting, bandwidth-throttling, bandwidth regulation) fatta **direttamente a livello kernel,** attraverso:
  - ✔ /lib/modules/[...] /netfilter/ipt_BCROUTER.ko – *kernel-space*
  - ✔ /lib/iptables/libipt_BCROUTER.so – *User-space*
- Una serie di tools che:
  - ✔ semplificano il dialogo con tale modulo, anche e soprattutto per la relativa configurazione;
  - ✔ gestiscono l'autenticazione dell'utente (Captive-Portal)

Per tutto il resto, BCrouter usa componenti STANDARD: interfacce, VLAN, DHCP...

BCROUTER *(...in breve)*

Molto sommariamente, un'ipotesi di impiego in una LAN particolarmente estesa (**> 1000 utenti simultanei**) è la seguente:



BCROUTER *(...in breve)*

Sulle performance di BCrouter, l riquadri riportano due slide preparate nel 2007 da **DIRK Jenssen**, lo sviluppatore principale (con cui siamo in contatto da tempo)

BCrouter: Performance

ICT Infrastructuur

- In use for more than 2 years on Kotnet
  - >45099 users in BCrouter database
  - >113420 IP addresses in BCrouter database
  - >500 Mbits bandwidth peak (30 min average)
  - >140 network segments (140 VLAN's)
- 1 Active server (with hot standby)
  - Dual Xeon 3,2Ghz
  - 1 Gigabyte RAM
  - Debian Linux (2.6 kernel)
- Peak CPU Load
  - 45% CPU total
    - 85% Linux general routing code
    - 15% BCrouter code
  - 430 Mbytes RAM in use for entire syste

K.U.LEUVEN – ICTI Netwerk

Ma allora perché nessuno lo usa?

Solution: integration

- Scalability
  - BCrouter server
    - Supports virtual unlimited users
      - Tested up to 50 000 users (1 Gigabyte RAM)
      - Handles up to 60 000 login/logout operations per second
    - Supports virtual unlimited IP addresses
      - Tested up to 200 000 IP's (1 Gigabyte RAM)
    - Supports up to 300 000 packets/sec (1.5 Gigabit)
      - Dual Xeon 3.6Ghz

BCROUTER *(...in breve)*

- La versione di BCrouter attualmente in produzione per KOTNET utilizza alcuni componenti il cui codice NON può essere "liberato" (es.: autenticazione);
- Tali componenti sono stati in gran parte riscritti (...unitamente ad una parte del "core" di BCrouter -**NetFlow V9 con username, Radius, NAT, etc.**).

Di fatto, ad oggi, BCRouter non è (ancora):
 # apt-get install bcrouter
o
 # yum install bcrouter

- La configurazione (di BCrouter e degli accessori – routing, captive portal, area di supporto, etc.) richiede una discreta quantità di tempo/uomo (...e di competenze).... esattamente come NOCAT (un po' meno...) e ChilliSpot (un po' piu'...)

A Chieti, BCrouter è "The way to go"

BCROUTER *(...in breve)*

Qualche riferimento:

- BCRouter:
  http://www.bcrouter.net

- "K.U. Leuven University":
  http://www.kuleuven.be/english/

- Canale IRC:
  #bcrouter su **irc.freenode.net**

Se qualcuno volesse "associarsi" a noi nel lavoro di Ricerca/Sviluppo/Utilizzo di BCrouter.... a noi (ed a tutti...) farebbe molto piacere!

BCrouter, ad oggi, È GIÀ UTILIZZABILE!!!

È tutto....

Domande?

Commenti, suggerimenti, critiche
(ed offerte di collaborazione...)
sono *__**più che bene accette**__*

Dr. Damiano Verzulli
damiano@verzulli.it

staff@noc.unich.it