

Manuale di FreePOPs

5th May 2004

Contents

1	Introduzione	2
1.1	Situazioni d'uso	2
1.2	Feature	2
1.3	Plugin	3
2	Storia	3
3	File di configurazione di FreePOPs	4
4	Parametri a riga di comando di FreePOPs	4
5	Configurazione del client email	5
6	Plugin	5
6.1	libero.lua	5
6.2	popforward.lua	6
6.3	aggregator.lua	6
6.4	flatnuke.lua	7
7	Creare un plugin	7
7.1	L'interfaccia tra il nucleo C ed un plugin	8
7.2	L'interfaccia tra un plugin e il nucleo C	9
8	Segnalare un bug	10
9	Autori	10
9.1	Sviluppatori	10
10	Ringraziamenti	11

1 Introduzione

FreePOPs e' un demone POP3 piu' un interprete LUA e alcune librerie extra per il parsing di HTTP e HTML. Il suo scopo principale e' tradurre richieste POP3 locali in azioni HTTP remote per le webmail supportate, ma in realta' e' piu' flessibile di cosi': per esempio esiste un plugin per leggere news da un sito web come se fossero messaggi in una mailbox. Si puo' facilmente estendere FreePOPs al volo, senza neanche farlo ripartire; si puo' aggiungere un plugin o modificarne uno esistente semplicemente cambiando uno script, dato che i plugin sono scritti in LUA e sono interpretati al volo.

1.1 Situazioni d'uso

FreePOPs puo' essere utile in molte situazioni, qui descriviamo le piu' ovvie:

- Siete dietro un firewall che chiude la porta 110 ma volete comunque leggere la posta elettronica e la webmail del vostro provider fa schifo.
- Il vostro mail provider non vi permette di accedere alla mailbox con il protocollo POP3 ma solo tramite il servizio di webmail.
- Preferite usare la vostra mailbox invece di sfogliare le news di un qualche sito.
- Dovete sviluppare un server POP3 in meno di una settimana e volete che sia ragionevolmente veloce e che non consumi molta memoria.
- Non siete hacker del C, ma volete trarre beneficio da un frontend ad un server POP3, veloce e scritto in C, ma non volete perdere un mese a scrivere il backend in C. LUA e' un linguaggio davvero semplice e leggero, una settimana e' abbastanza per impararlo e poterlo usare in modo produttivo.

1.2 Feature

FreePOPs e' l'unico software che conosciamo con queste feature:

- Server POP3 compliant con RFC (non con tutte le feature, ma compliant).
- Portabile (scritto in C e LUA il quale e' scritto in C, quindi tutto e' scritto nel linguaggio piu' portabile del mondo).
- Piccolo (in termini di risorse utilizzate) e ragionevolmente veloce.
- Estremamente estendibile al volo mediante un linguaggio semplice e potente.

- Piuttosto documentato.
- Rilasciato sotto la licenza GNU/GPL (questo significa che FreePOPs e' Software Libero).

1.3 Plugin

Questi sono i plugin correntemente inclusi in FreePOPs:

libero.lua Questo plugin supporta in pieno la webmail di www.libero.it per mailbox con domini come @libero.it, @iol.it, @inwind.it, @blu.it.

popforward.lua Questo e' un plugin usato principalmente per testare moduli di FreePOPs. Esso agisce come un forward POP3, semplicemente si comporta come mediatore tra voi ed un vero server POP3. Questo plugin ci ha permesso di testare FreePOPs senza avere alcun vero plugin gia' scritto. Potreste usarlo per mascherare un server POP3 con molti bug che puo' venire facilmente compromesso grazie a richieste malevole. Ovviamente suggeriamo di esaminare per bene questo plugin, e fare un po' di hacking per prevenire richieste malevole al vostro server.

aggregator.lua Molti siti forniscono un backend RSS per indicizzare le loro news. Questo plugin fa si' che RSS si comporti come una mailbox in cui potete trovare una mail per ogni news.

flatnuke.lua Questo e' un aggregator piu' potente per siti basati sul CMS Flat-Nuke¹, e permette il download dell'intero corpo delle news.

2 Storia

FreePOPs non nasce dal nulla. Un progetto simile (solo nella situazione d'uso principale) e' LiberoPOPs.

L'antenato di FreePOPs e' completamente scritto in C per ragioni poco interessanti. LiberoPOPs supporta "plugin" ma in maniera piu' statica e complessa. Il frontend al server POP3 potrebbe essere collegato ad un backend scritto in C, questo significa che dovrete ricompilare e far ripartire LiberoPOPs ogni volta che cambiate una riga in un plugin. Un altro punto interessante e' che LiberoPOPs era stato creato dal nulla in un tempo molto breve (dovete essere Italiani e usare un indirizzo di posta@libero.it per capire perche'), cio' vuol dire che era nato con molti bug e FIX-ME nel codice.

¹Vedete [HTTP://flatnuke.sourceforge.net](http://flatnuke.sourceforge.net) per l'homepage del progetto

Il progetto LiberoPOPs ebbe un rapido successo, perché tutti ne avevano bisogno, quindi avevamo molti utenti. Nella filosofia della comunità open-source (e anche di Linux) devi rilasciare il software frequentemente, e questo è ciò che facevamo: rilasciavamo nuove versioni ogni due giorni. Non avevamo a che fare con utenti Unix, né hacker, ma per la maggior parte utenti Win32. Ad un certo punto capimmo che questi erano pigri/stufi di aggiornare il software ogni due giorni. Il brutto mondo Win insegna che il software si auto-aggiorna, si auto-installa e probabilmente si auto-scrive.

Cercammo di risolvere il problema tirando fuori dal motore in C la maggior parte del codice che cambiava più spesso, ma questo era molto difficile visto che il C non è pensato per questo genere di cose. Una volta che LiberoPOPs si fu stabilizzato iniziammo a pensare a come risolvere meglio la cosa.

Un linguaggio di scripting/interpretato sembra una buona scelta e dopo una lunga ricerca in rete e nei newsgroup universitari trovai LUA.. Questo non è il luogo per dire al mondo quanto sia bello questo linguaggio quindi non ne parlerò oltre qui. Integrare l'interprete LUA in LiberoPOPs non fu così difficile e FreePOPs ne è il risultato. Ora è davvero più facile scrivere/testare un plugin e (anche se non è ancora implementato) un sistema di auto-aggiornamento è molto facile da scrivere dato che non c'è bisogno di ricompilare il nucleo C in quasi nessun caso.

3 File di configurazione di FreePOPs

FreePOPs non ha bisogno di una vera configurazione. La maggior parte degli utenti non dovrebbe modificare il file di configurazione. Se siete sviluppatori o utenti curiosi il file di configurazione è `config.lua`, che si trova nella directory del programma sotto win32 o in `/etc/freepops/` in ambiente posix.

Più avanti vedremo come i plugin sono associati al dominio di un indirizzo di posta, e alcuni di questi plugin hanno alias per altri domini per rendere più facile la raccolta di news da alcuni siti. Leggete la documentazione dei plugin per maggiori informazioni su di essi, e magari inviate una mail con il vostro nuovo alias se volete che venga integrato nella prossima versione di FreePOPs.

4 Parametri a riga di comando di FreePOPs

La vera configurazione di FreePOPs viene impostata tramite argomenti a riga di comando. Questi sono descritti in dettaglio nelle pagine del man in ambienti Unix, qui presentiamo solo i più utili:

-p <port>, -port <port> Per default FreePOPs fa bind sulla porta 2000. Per modificare questo comportamento basta usare questo switch.

- P <host>:<port>, -proxy <host>:<port>** Per dire a FreePOPs quale e' il vostro proxy HTTP.
- v, -verbose, -w, -veryverbose** Questo dice a FreePOPs di loggare alcune informazioni utili per riportare bug.
- t <num>, -threads <num>** FreePOPs puo' gestire connessioni multiple, fino a num. Il default e' 5.

In ambienti posix come Debian GNU/Linux potete avviare FreePOPs al boot come servizio standard. In questo caso gli switch a riga di comando sono memorizzati in `/etc/default/freepops`, in alcuni sistemi basati su rpm dovreste trovare lo stesso file con nome `/etc/sysconfig/freepops`.

5 Configurazione del client email

Per configurare il client email dovete cambiare le impostazioni del server POP3. Solitamente dovreste usare localhost come nome del server POP3 e 2000 come porta. Nel caso in cui installiate FreePOPs in un altro computer della vostra LAN, dovreste usare il nome di quell'host invece di localhost, mentre nel caso in cui abbiate cambiato la porta di default con lo switch `-p` dovreste immettere la stessa porta anche nel client email.. Dovete sempre usare come nome utente l'indirizzo di posta completo, per esempio `qualcosa@libero.it` invece che solo `qualcosa`. Questo e' perche' FreePOPs sceglie il plugin da caricare guardando al nome utente che deve quindi contenere tutte le informazioni. Piu' sotto presentiamo tutti i plugin e i loro domini associati.

6 Plugin

Qui diamo una descrizione dettagliata di ogni plugin.

6.1 libero.lua

Questo plugin vi permette di leggere le mail che avete in una `mailbox@libero.it`, `@iol.it`, `@inwind.it` e `@blu.it`. Cio' significa che potete ancora usare il vostro mail reader preferito invece di usare la webmail. Questo plugin agisce come un browser che sfoglia il vostro account webmail e lo fa apparire come un server POP3. Per maggiori informazioni su questo plugin potete guardare il sito web di LiberoPOPs (antenato di FreePOPs) <http://liberopops.sourceforge.net>

Per usare questo plugin dovete usare il vostro indirizzo email completo come username e la vostra password reale come password.

6.2 **popforward.lua**

Questo plugin e' stato sviluppato per testare FreePOPs prima che fossero scritti altri veri plugin. Esso semplicemente inoltra richieste locali verso un vero server POP3. Puo' essere usato per mascherare un server POP3 che ha dei bug, ma se pensate di averne bisogno dovrete esaminare attentamente il codice del plugin e aggiungere controlli per migliorare il rilevamento/la prevenzione di richieste malevole, dato che il plugin stesso non e' nato con in mente la sicurezza.

Per usare questo plugin dovete modificare il file `config.lua`. Questo perche' non possiamo aggiungere tutti i server POP3 esistenti al file :) Il plugin richiede due argomenti, l'host POP3 e la porta (di solito 110) su cui il server e' in ascolto. Questo e' un esempio di una riga di configurazione per questo plugin, in cui ogni indirizzo email del dominio `@virgilio.it` sono inoltrati a `in.virgilio.it:110`:

```
-- popforward plugin
freepops.MODULES_MAP["virgilio.it"] = {
    name="popforward.lua",
    args={
        port=110,
        host="in.virgilio.it"
    }
}
```

6.3 **aggregator.lua**

Solitamente potete trarre beneficio dal formato RSS del W3C quando leggete news da qualche sito web. Il file RSS indicizza le news, fornendo un link verso di esse. Questo plugin puo' far si' che il vostro client di posta veda il file RSS come una mailbox da cui potete scaricare ogni news come se fosse una mail. L'unica limitazione e' che questo plugin puo' prelevare solo un sunto delle news piu' il link alle news.

Per usare questo plugin dovete usare un nome utente casuale con il suffisso `@aggregator` (es.: `foo@aggregator`) e come password l'URL del file RSS (es.: `http://www.securityfocus.com/rss/vulnerabilities.xml`). Per comodita' abbiamo aggiunto per voi alcuni alias. Questo significa che non dovrete cercare a mano l'URL del file RSS. Abbiamo aggiunto alcuni domini, per esempio `@securityfocus.com`, che possono essere usati per sfruttare direttamente il plugin aggregator con questi siti web. Per usare questi alias dovrete usare un nome utente nella forma `qualcosa@aggregatordomain` e una password a caso. Questa e' la lista di alias per il plugin aggregator.

aggregatordomain	descrizione
freepops.rss.en	http://freepops.sourceforge.net/ news (English)
freepops.rss.it	http://freepops.sourceforge.net/ news (Italian)
flatnuke.sf.net	http://flatnuke.sourceforge.net/ news (Italian)
ziobudda.net	http://ziobudda.net/ news (both Italian and English)
punto-informatico.it	http://punto-informatico.it/ news (Italian)
linuxdevices.com	http://linuxdevices.com/ news (English)
securityfocus.com	http://www.securityfocus.com/ new vulnerabilities (English)

6.4 flatnuke.lua

Questo plugin e' un plugin aggregator specializzato nei siti web fatti con il CMS FlatNuke², o altri siti che usano lo stesso formato delle news come il sito di FreePOPs. Dato che in un sito FlatNuke le news sono memorizzate in semplici file xml questo plugin e' in grado di prelevare tutte le news, non solo le intestazioni come il plugin aggregator. Cio' e' molto utile se non vuoi sfogliare l'intero sito web per leggere le news.

Per usare questo plugin dovete avere un nome utente con il dominio @flatnuke (es.: qualcosa@flatnuke) e l'URL di una homepage flatnuke come password (es.: <http://flatnuke.sourceforge.net/>, non c'e' bisogno di URL di file RSS visto che FlatNuke mette gli RSS in una posizione nota e fissata. Ci sono alcuni alias per siti FlatNuke, vedi la documentazione del plugin aggregator per sapere cosa significa):

aggregatordomain	descrizione
freepops.en	http://freepops.sourceforge.net/ full news (English)
freepops.it	http://freepops.sourceforge.net/ full news (Italian)
flatnuke.it	http://flatnuke.sourceforge.net/ full news (Italian)

7 Creare un plugin

Il modo migliore di farlo e' leggere attentamente il file libero.lua, che e' un esempio molto semplice ma anche molto commentato di plugin per webmail. Poi dovrete copiare il file skeleton.lua e farci un po' di hacking. Ricordate di modificare config.lua per far si' che FreePOPs associ i giusti domini di indirizzi email al vostro nuovo plugin. Un esempio piu' semplice e' flatnuke.lua che e' un plugin per web news.

²[HTTP://flatnuke.sourceforge.net](http://flatnuke.sourceforge.net)

7.1 L'interfaccia tra il nucleo C ed un plugin

Come abbiamo spiegato prima il frontend POP3 in C deve essere collegato ad un backend in LUA. L'interfaccia e' molto semplice se conoscete il protocollo POP3. Qui riassumiamo brevemente il significato, ma la RFC 1939 (inclusa nella directory `doc/` della distribuzione dei sorgenti) e' molto breve e facile da leggere. Come il vostro intuito dovrebbe suggerirvi il client POP3 puo' richiedere che il server POP3 conosca qualcosa delle mail che sono nella mailbox e prima o poi prelevare/cancellare dei messaggi. E questo e' esattamente cio' che fa.

Il backend deve implementare tutti i comandi POP3 (come USER, PASS, RETR, DELE, QUIT, LIST, ...) e deve restituire al frontend il risultato. Diamo un semplice esempio di una sessione POP3 dalla RFC:

```
1  S: <wait for connection on TCP port 110>
2  C: <open connection>
3  S:   +OK POP3 server
4  C:   USER linux@kernel.org
5  S:   +OK now insert the password
6  C:   PASS gpl
7  S:   +OK linux's maildrop has 2 messages (320 octets)
8  C:   STAT
9  S:   +OK 1 320
10 C:   LIST
11 S:   +OK 2 messages (320 octets)
12 S:   1 320
13 S:   .
14 C:   RETR 1
15 S:   +OK 120 octets
16 S:   <the POP3 server sends message 1>
17 S:   .
18 C:   DELE 1
19 S:   +OK message 1 deleted
20 C:   QUIT
21 S:   +OK dewey POP3 server signing off (maildrop empty)
22 C:   <close connection>
23 S:   <wait for next connection>
```

In questa sessione il backend verra' chiamato per le righe 4, 6, 8, 10, 14, 18, 20 (tutte le righe C:) e rispettivamente le funzioni che implementano i comandi POP3 verranno chiamate in questo modo

```
user(p, "linux@kernel.org")
pass(p, "gpl")
stat(p)
list_all(p)
```



```

retr(p,1)
dele(p,1)
quit_update(p)

```

Piu' tardi chiariremo cos'e' p. Speriamo di toglierlo e renderlo implicito per completa trasparenza. E' facile capire che c'e' un mapping 1-1 tra i comandi POP3 e le chiamate a funzione del plugin. Potete vedere un plugin come l'implementazione dell'interfaccia POP3.

7.2 L'interfaccia tra un plugin e il nucleo C

Prendiamo in esame la chiamata a `pass(p, "linux@kernel.org")`. Qui il plugin dovrebbe autenticare l'utente (se c'e' un qualche tipo di autenticazione) e informare il nucleo C del risultato. Per ottenere questo ogni funzione dei plugin deve restituire un flag di errore, per essere piu' precisi uno di questi errori:

Code	Significato
POPSERVER_ERR_OK	Nessun errore
POPSERVER_ERR_NETWORK	Errore di rete
POPSERVER_ERR_AUTH	Autenticazione fallita
POPSERVER_ERR_INTERNAL	Errore interno, segnalate il bug
POPSERVER_ERR_NOMSG	Il numero del messaggio e' fuori range
POPSERVER_ERR_LOCKED	Mailbox bloccata da altre sessioni
POPSERVER_ERR_EOF	Fine trasmissione, usata nel <code>popserver_callback</code>
POPSERVER_ERR_TOOFAST	Non e' possibile riconnettersi al server ora, attendere e riprovare
POPSERVER_ERR_UNKNOWN	Non ho idea di che errore ho trovato

Nel nostro caso i codici d'errore piu' appropriati sono `POPSERVER_ERR_AUTH` e `POPSERVER_ERR_OK`. Questo e' un caso semplice, in cui un codice d'errore e' abbastanza. Ora analizziamo il caso piu' complesso della chiamata a `list_all(p)`. Qui dobbiamo restituire un codice d'errore come prima, ma dobbiamo anche informare il nucleo C della grandezza di tutti i messaggi nella mailbox. Qui abbiamo bisogno del parametro p passato ad ogni funzione del plugin (notate che tale parametro potra' divenire implicito in futuro). p indica la struttura dati che il C si aspetta venga riempita chiamando funzioni appropriate come `set_mailmessage_size(p, num, size)` dove num e' il numero del messaggio e size e' la grandezza in byte. Solitamente e' molto comune mettere insieme piu' funzioni. Per esempio quando guardate la pagina di una webmail con la lista di messaggi conoscete il numero dei messaggi, la loro grandezza e lo UIDL cosi' che potete riempire la struttura dati p con tutte le informazioni per LIST, STAT, UIDL.

L'ultimo caso che esaminiamo e' `retr(p,num,data)`. Poiche' un messaggio di posta puo' essere molto grande, non e' un modo elegante di scaricare l'intero messaggio senza far si' che il client di posta si lamenti per la morte del server. La soluzione e' usare un callback. Ogni volta che un plugin ha dei dati da mandare al client dovrebbe chiamare la `popserver_callback(buffer,data)`. `data` e' una struttura opaca che il popserver necessita per compiere il suo lavoro (notate che questo parametro potra' venire rimosso per semplicita'). In alcuni casi, per esempio se sapete che il messaggio e' piccolo o state lavorando su una rete veloce, potete prelevare l'intero messaggio e mandarlo, ma ricordate che questo consuma piu' memoria.

8 Segnalare un bug

Quando avete problemi o pensate di avere trovato un bug, vi preghiamo di seguire alla lettera questo *iter*:

1. Aggiornate alla versione piu' recente di FreePOPs.
2. Cercate di riprodurre il bug, se questo non e' facilmente riproducibile siamo sfortunati. Si puo' ancora tentare qualcosa, se il software e' andato in crash potreste compilarlo dai sorgenti, installare valgrind, lanciare `freepopd` con valgrind e sperare che i messaggi d'errore siano interessanti.
3. Pulite i file di log
4. Lanciate FreePOPs con lo switch `-w`
5. Riproducete il bug
6. Inviare agli sviluppatori il log, piu' ogni altra informazione utile come che tipo di sistema avete e come riprodurre il bug.

9 Autori

Questo manuale e' stato scritto da Enrico Tassi <gareuselesinge@users.sourceforge.net> and rivisto e tradotto da Nicola Cocchiaro <ncocchiaro@users.sourceforge.net>

9.1 Sviluppatori

FreePOPs e' sviluppato da:

- Enico Tassi <gareuselesinge@users.sourceforge.net>
- Alessio Caprari <alessiofender@users.sourceforge.net>

- Nicola Cocchiaro <ncocchiaro@users.sourceforge.net>
- Simone Vellei <simone_vellei@users.sourceforge.net>

LiberoPOPs e' sviluppato da:

- Enrico Tassi <gareuselesinge@users.sourceforge.net>
- Alessio Caprari <alessiofender@users.sourceforge.net>
- Nicola Cocchiaro <ncocchiaro@users.sourceforge.net>
- Simone Vellei <simone_vellei@users.sourceforge.net>
- Giacomo Tenaglia <sonicsmith@users.sourceforge.net>

10 Ringraziamenti

Ringraziamenti speciali vanno agli utenti che hanno testato il software, agli hacker che hanno reso possibile avere un ambiente di sviluppo affidabile e libero come il sistema Debian GNU/Linux.