Intromettersi fra un applicazione ed il server che usa per ricevere dati

aprile 20th, 2023

Non so se avete letto l'articolo SOLUZIONE: Hacking This Site – Application – Application Challenge 2 in cui spiegavo come crakkare un programma sniffando i pacchetti trasmessi e ricevuti dallo stesso.
Per riassumere il programma ci avrebbe dato una password, a patto che il codice di licenza inserito, a noi completamente sconosciuto, fosse stato equivalente a uno tra quelli rilasciati da un server.
In questo articolo vi spiegherò come interporvi fra voi ed il server.
Sembra quasi un Attacco man in the middle con la differenza che una delle due parti viene completamente tagliata fuori, in questo caso il server.
Il software penserà di comunicare con il server ma in realtà starà comunicando con noi.
Questo sarà possibile, grazie ad un dirottamento della connessione. Il dirottamento consiste nel blindare qualsiasi connessione verso il server da parte del nostro pc, e mettere in piedi un altro server che lo andrà a simulare e accetterà le connessioni da parte dell'applicazione.
Dall'altra parte invece, il server non riceverà alcuna comunicazione dal software che vogliamo crakkare.
Detto questo direi che possiamo iniziare, ma prima penso che sia fondamentale che vi leggiate l'articolo SOLUZIONE: Hacking This Site – Application – Application Challenge 2.
Premettendo che avete già letto l'articolo precedente, partiamo facendo in modo tale che il nostro computer non possa comunicare con il server hackthissite.org

Se andate sul sito Hack This Site vedrete che tutto funziona correttamente.
Potete consultare il sito senza problemi.
Per blindare la connessione, eseguiamo “Blocco note” come amministratore ed andiamo ad aprire il file seguente.

```
C:\Windows\System32\drivers\etc\hosts
```

Quindi andiamo ad aggiungere in coda al file il comando visualizzato sotto.

```
127.0.0.1       hackthissite.org
```

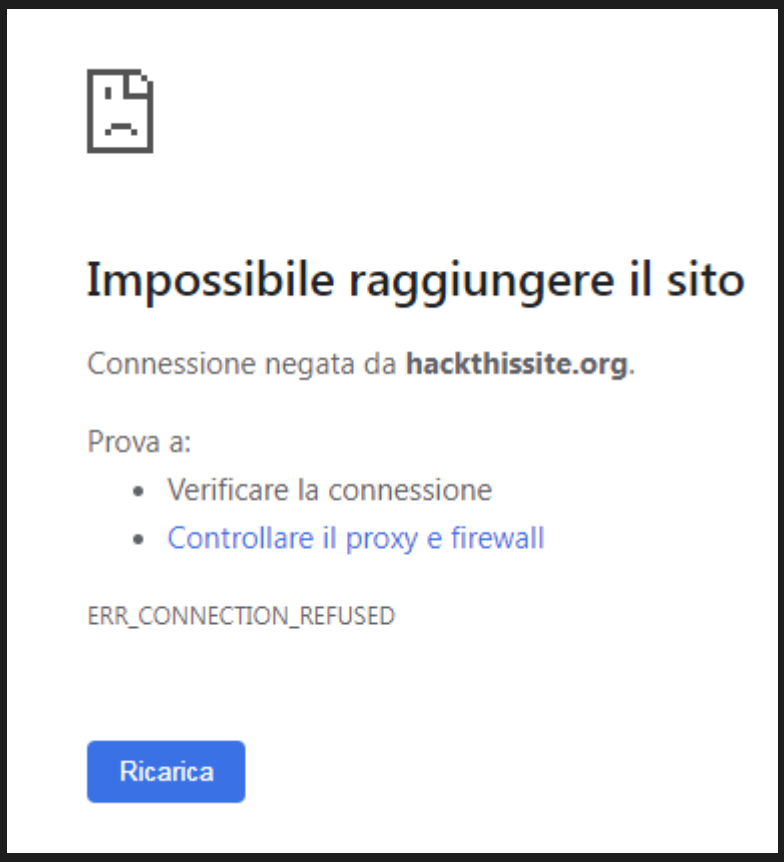
Fra l’ip 127.0.0.1 che rappresenta il nostro ip locale e hackthissite.org c’è uno spazio derivante da una tabulazione. In pratica dovrete premere solo il tasto TAB.
Di seguito vi riporto il contenuto del mio file hosts modificato.
Il file è composto esclusivamente da commenti che spiegano il suo funzionamento ed il record appena aggiunto.
I commenti sono preceduti dal carattere #.

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host
name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host
```

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

127.0.0.1       hackthissite.org
```

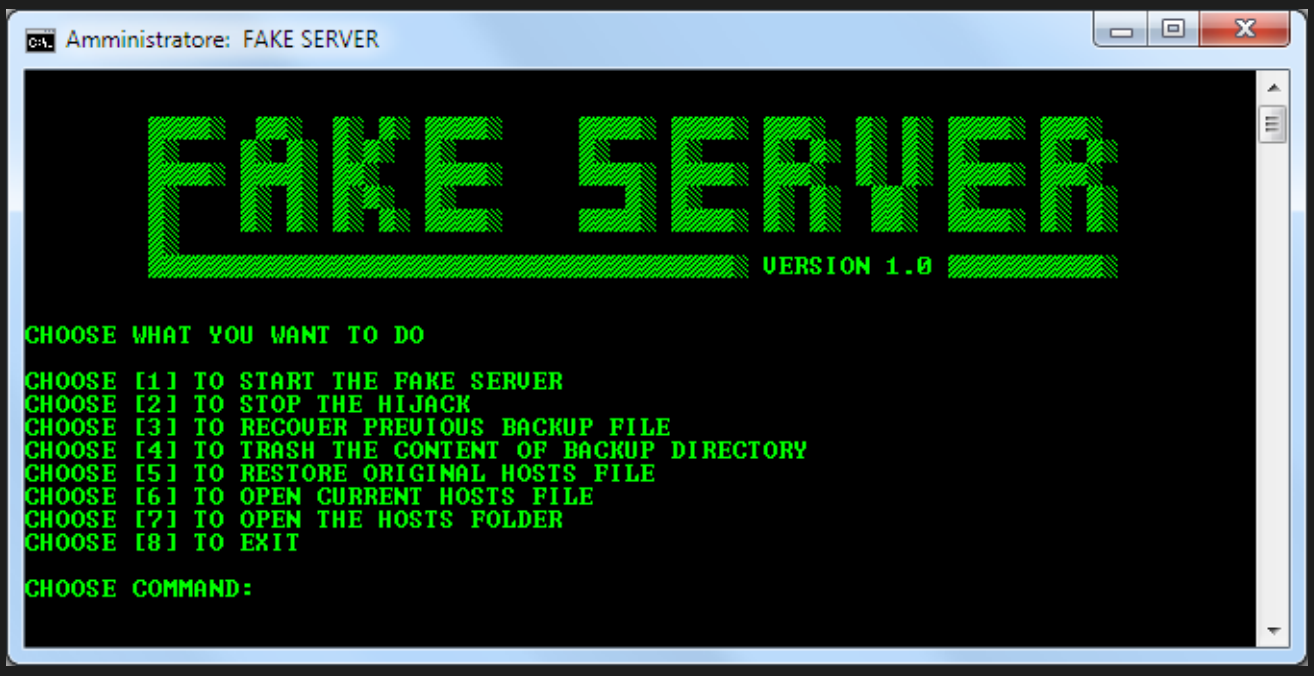
Questo comando dice che, se dal nostro pc dovesse partire qualsiasi connessione verso il dominio hackthissite.org, questa dovrà essere dirottata al nostro ip locale 127.0.0.1
Grazie a questo trucco è possibile fare tantissimi scherzetti, ma al tempo stesso anche proteggere il pc dalla connessione verso server in grado di piegarci il computer con programmi dannosi.
Se avete applicato la modifica, e provate ad andare sul sito Hack This Site otterete questo.



Niente paura, perchè vi basterà eliminare il record scritto prima sul file hosts per ritornare alla condizione precedente e togliere così il dirottamento.
A questo punto dobbiamo creare un server che rimanga in attesa di una connessione da parte di qualsiasi software che cerchi di collegarsi a hackthissite.org
Qualsiasi software presente nel nostro computer riuscirà a comunicare con

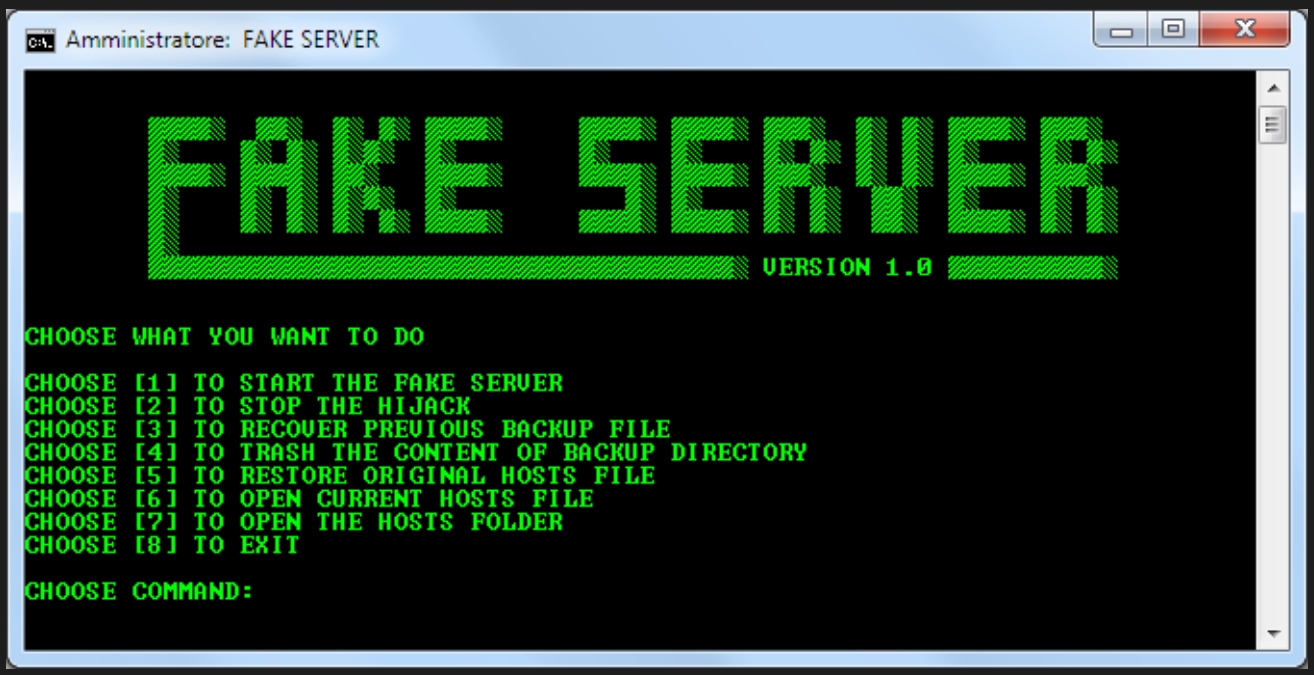
gli eventuali server, tranne quelli che proveranno a collegarsi al server che risponde al dominio hackthissite.org
In questo caso la connessione verrà dirottata verso il nostro ip locale 127.0.0.1
Questo sistema può essere utile anche per monitorare l'attività di un programma e sniffare le sue richieste, anche se vi consiglio di usare Wireshark che è sicuramente più avanzato.
Potete utilizzare questo sistema anche per creare una sorta di proxy.
Il client prova a comunicare con un server ma la connessione viene dirottata al nostro proxy. Il proxy che farà da intermediario, poi decide se sbloccare la connessione inoltrando le richieste al server e restituendo i dati al client, o andare a blindare la connessione.
Ovviamente si tratta solo di qualche esempio.
Ritorniamo alla creazione del server.
Per mettere in piedi un server per questo giochetto non è necessario diventare pazzi. Io in questi casi uso NETCAT. Meglio conosciuto come il coltellino svizzero delle reti. Permette di mettere in piedi un client o un server in pochissimi secondi. Racchiude moltissime funzioni e direi che è uno strumento indispensabile.
Lo potete scaricare dal sito NMAP.ORG
Per rendervi ancora più semplice questa esperienza, ho deciso di sviluppare un file BATCH che potrete scaricare facendo click sull'immagine seguente.

FAKE SERVER



Il BATCH l'ho chiamato FAKE SERVER.
Ho adottato una struttura rivoluzionaria del codice sorgente, aggiungendo anche un tocco di grafica in vecchio stile.
Vi ho illustrato tutte le varie procedure da utilizzare per mettere in piedi questo sistema, ma non siete costretti ad eseguirle manualmente. Ci penserà FAKE SERVER a fare tutto al posto vostro.
Piuttosto che spiegarvi ogni singolo comando che va a comporre FAKE SERVER vi illustro quello che fa, ma non prima di avervi spiegato di cosa abbiamo bisogno.
Facciamo un riassunto.
Come avevo scritto, nel pc dobbiamo dirottare qualsiasi connessione verso hackthissite.org a 127.0.0.1
Abbiamo bisogno di creare un server che vada a simulare il server che risponde al dominio hackthissite.org
Successivamente, una volta che il nostro server è in esecuzione, non dobbiamo fare altro che avviare l'applicazione Application Challenge 2.
In seguito all'immissione di un product key, Application Challenge 2 proverà a collegarsi a hackthissite.org per scaricare i product key validi e confrontarli con quello che abbiamo inserito. In realtà si collegherà al nostro server che restituirà quello che avrebbe restituito hackthissite.org ma con una piccola modifica.
Il FAKE SERVER infatti trasmetterà dei product key personalizzati che saranno reputati comunque validi dal programma Application Challenge 2. Il programma quindi ci restituirà la password.
FAKE SERVER, può essere utilizzato per crakkare tutti i programmi che hanno un comportamento simile ad Application Challenge 2.
Ovviamente non essendo un server professionale ha qualche limitazione.
Ogni tanto i programmi, quando si collegano ai server per inviare e ricevere pacchetti, spesso richiedono un file e in altri casi richiedono il download di più file.
Per far scaricare al software da attaccare più file modificati, è necessario dotarsi di un applicazione che ci consenta di creare un server professionale.
Magari affronteremo il discorso in seguito.

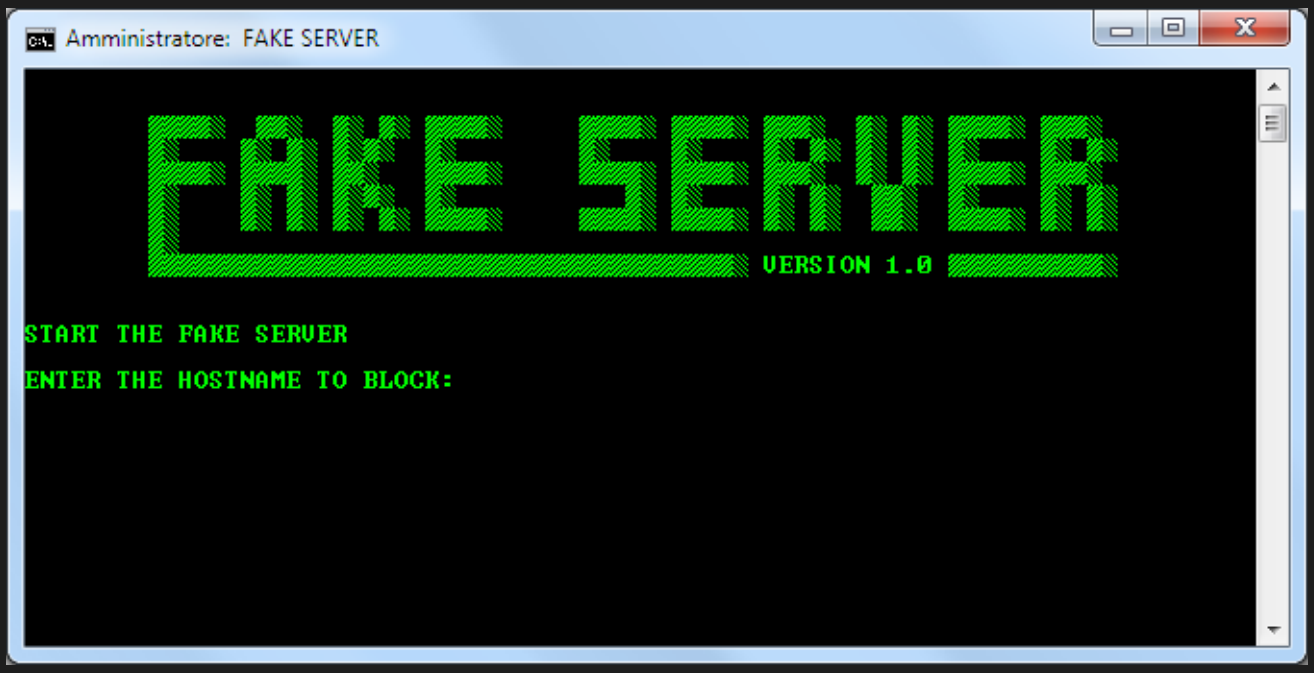
Per utilizzare FAKE SERVER è inoltre fondamentale capire prima che tipo di comunicazione avviene fra l'applicazione che vogliamo crakkare ed il server, in quanto poi dovremmo sapere quali dati dargli in pasto e come possiamo modificarli.
Adesso vi spiego come funziona FAKE SERVER.
Una volta scaricato, decomprimete il file compresso ed entrate dentro la cartella FAKE SERVER.
Avviate FAKE SERVER facendo doppio click sul file `SERVER.BAT`
FAKE SERVER vi presenterà un menu.



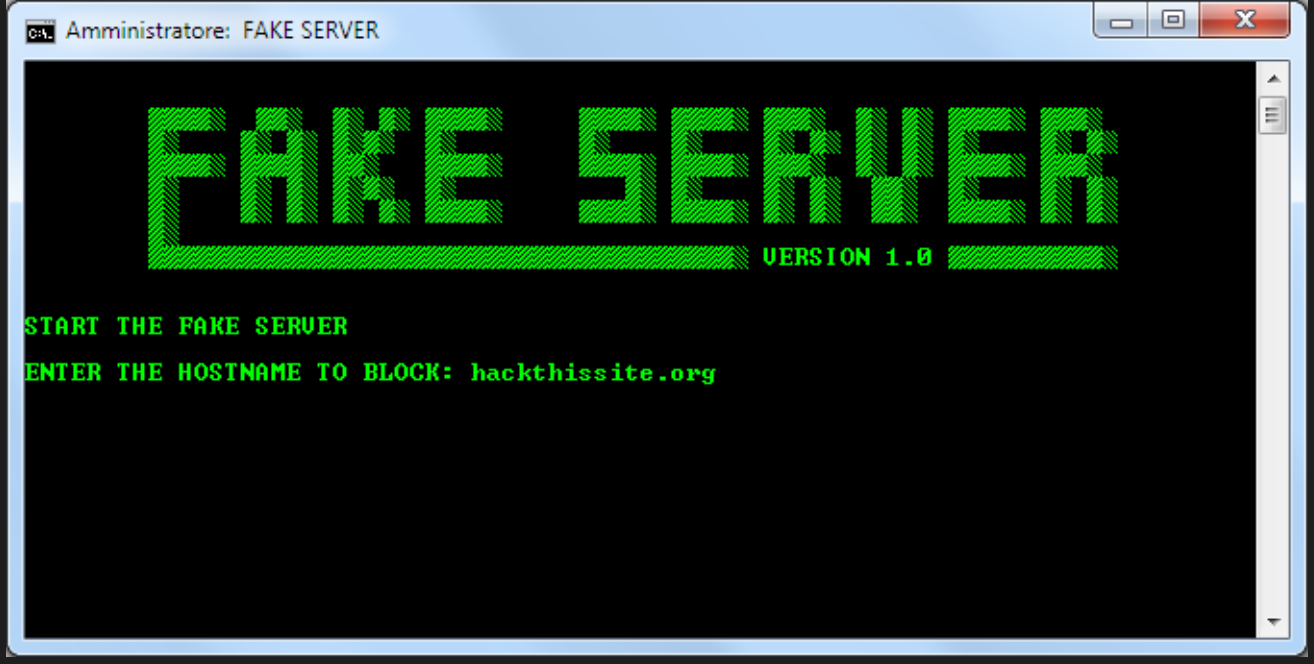
FAKE SERVER è stato sviluppato in inglese. Ad ogni modo la tabella riepilogativa presentata di seguito, dovrebbe aiutarvi a comprendere meglio la funzione associata ad ogni tasto.

TASTO	FUNZIONE
1	AVVIA FAKE SERVER
2	INTERROMPI IL DIROTTAMENTO
3	RIPRISTINA IL BACKUP DEL FILE hosts
4	ELIMINA TUTTI I FILE DI BACKUP DEL FILE hosts
5	RIPRISTINA IL FILE hosts ORIGINALE
6	APRI IL FILE hosts ATTUALMENTE USATO DAL SISTEMA
7	APRI LA CARTELLA CHE CONTIENE IL FILE hosts ATTUALMENTE UTILIZZATO DAL SISTEMA
8	ESCI DA FAKE SERVER

Per avviare il server che sarà utilizzato per crakkare l'applicazione, premete il pulsante `1` sulla vostra tastiera. In alternativa, potete premere il tasto `TAB` varie volte, fino a quando non vi sarà proposto il numero associato alla funzione che desiderate eseguire.
Scegliendo la funzione associata al tasto `1`, vi sarà mostrata la schermata seguente.



Vi verrà chiesto di specificare l'hostname al quale risponde il server che dovremmo simulare. Quello al quale l'applicazione che stiamo cercando di crakkare tenterà di collegarsi.
Avete due possibilità per farlo. Premere `TAB` fino a quando non compare il dominio desiderato, oppure digitare direttamente `hackthissite.org` ed in seguito premete `INVIO`.
FAKE SERVER memorizzerà tutti gli hostname digitati. Li salverà all'interno della cartella `INPUT\HOSTNAME` e li potrete sfogliare durante questo processo premendo il tasto `TAB`.



Successivamente vi verrà chiesto di specificare una porta compresa tra `1` e `65535`. Per maggiori informazioni vi suggerisco di consultare la pagina Porte TCP e UDP standard.

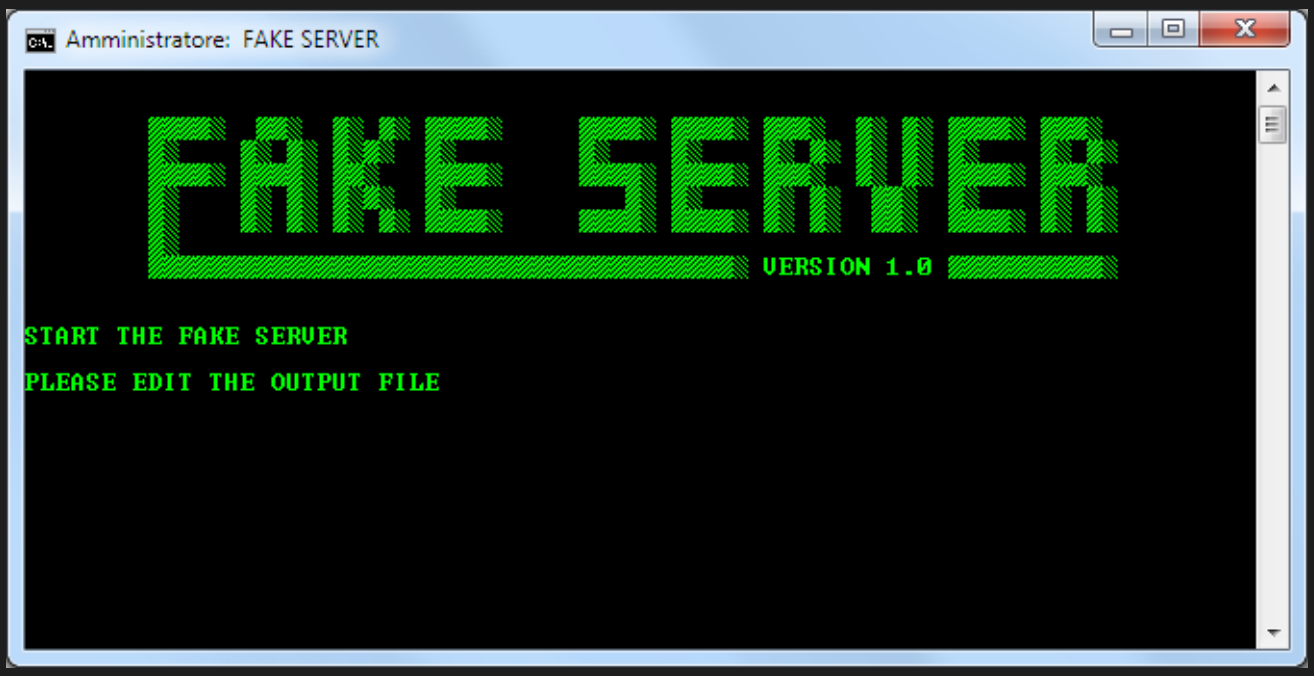
Naturalmente la porta dovrà essere quella che viene utilizzata dall'applicazione che vogliamo crakkare, per dialogare con il proprio server. La otterrete durante l'analisi con il software di packet sniffing, come Wireshark.

Se non viene specificata alcuna porta e premete `INVIO`, di default viene utilizzata la porta `80`, che è la porta abitualmente utilizzata per il protocollo HTTP.

In alternativa potrete digitare manualmente il numero della porta che sarà salvata all'interno della cartella `INPUT\PORT`.

In seguito, durante questa richiesta, potete sfogliare le varie porte utilizzate in precedenza agendo sul tasto TAB, o in alternativa potete digitare il numero di porta manualmente. Al momento usando il tasto TAB potrete selezionare una fra le porte che reputo più importanti.

Successivamente se non verrà trovato alcun file OUTPUT.TXT ne sarà creato uno e vi sarà proposto di modificarlo. In alternativa vi sarà proposta la modifica del file OUTPUT.TXT già presente all'interno della cartella di FAKE SERVER.



Ricordatevi di salvare il file OUTPUT.TXT dopo la modifica.
Il file OUTPUT.TXT in seguito verrà dato in pasto alla prima applicazione che si collegherà a FAKE SERVER.
Almeno con Application Challenge 2, vi potete permettere di modificare il

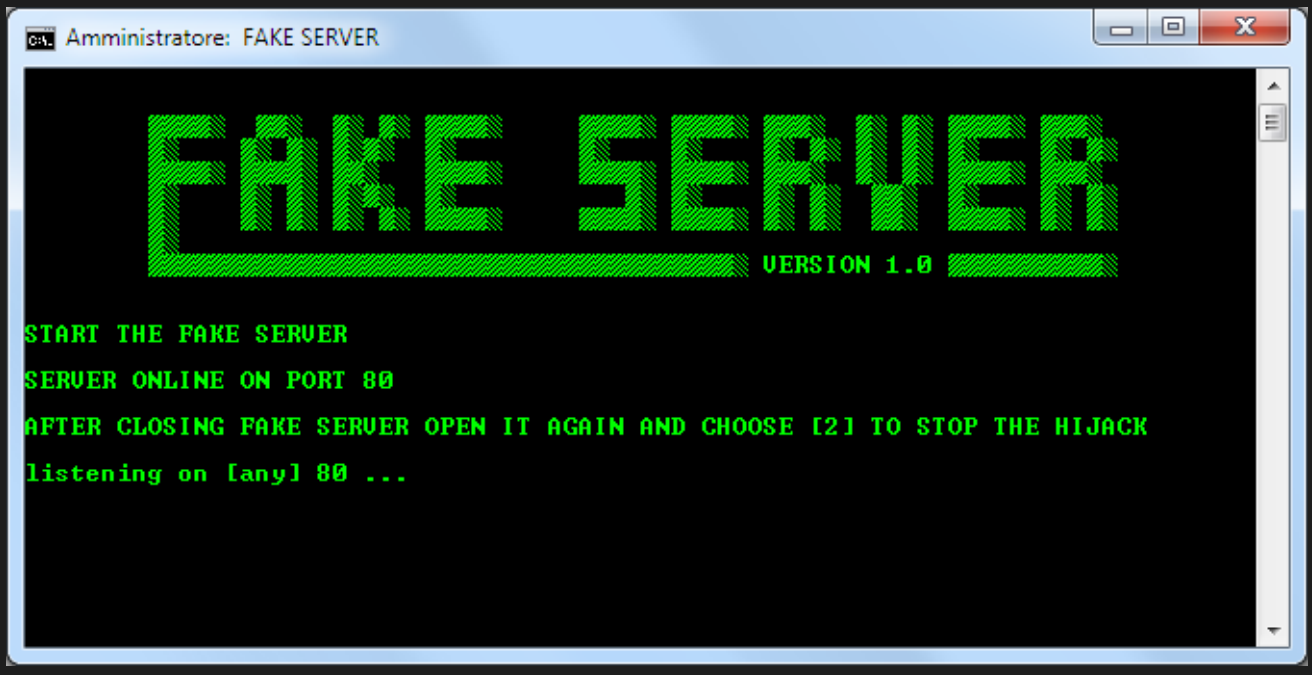
file `OUTPUT.TXT` in tempo reale per fare delle prove, in quanto Application Challenge 2 richiederà la ricezione dei product key validi ogni volta che faremo click sul pulsante “Authenticate”. FAKE SERVER in quel momento trasmetterà i dati richiesti e contenuti nel file `OUTPUT.TXT`
Di seguito vi mostro i dati presenti all’interno del file `OUTPUT.TXT` che potete trovare nella cartella di FAKE SERVER.
Al suo interno troverete già i dati necessari per crakkare Application Challenge 2.
Per darlo in pasto ad un software diverso, potete modificare OUTPUT.TXT come preferite. Ovviamente dovrà essere modificato in modo corretto. Dovrete inserire i comandi personalizzati da dare in pasto al software da crakkare, ma sulla base dei pacchetti analizzati precedentemente usando software come Wireshark od un altro packet sniffer.

```
HTTP/1.1 200 OK
Date: Mon, 17 Apr 2023 22:19:59 GMT
Upgrade: h2,h2c
Connection: Upgrade, close
Vary: Accept-Encoding
Content-Length: 299
Content-Type: text/html
Content-Language: en
Server: HackThisSite
Access-Control-Allow-Origin: *
Content-Security-Policy: child-src 'self' hackthissite.org
*.hackthissite.org htscdn.org *.htscdn.org discord.com; form-action
'self' hackthissite.org *.hackthissite.org htscdn.org *.htscdn.org;
upgrade-insecure-requests; report-uri https://hackthissite.report-
uri.com/r/d/csp/enforce
Referrer-Policy: origin-when-cross-origin
X-XSS-Protection: 0
Feature-Policy: fullscreen *
Public-Key-Pins-Report-Only: pin-
sha256="YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg="; pin-
sha256="Vjs8r4z+80wjNcr1YKepWQboSIRi63WsWXhIMN+eWys="; max-
age=2592000; includeSubDomains; report-
uri="https://hackthissite.report-uri.com/r/d/hpkp/reportOnly"
Strict-Transport-Security: max-age=31536000; includeSubDomains;
preload
```

```
Report-To: {"group":"default","max_age":31536000,"endpoints":
[{"url":"https://hackthissite.report-
uri.com/a/d/g"}],"include_subdomains":true}
NEL:
{"report_to":"default","max_age":31536000,"include_subdomains":true,"s
uccess_fraction":0.0,"failure_fraction":0.1}

11111-11111-11111-11111
22222-22222-22222-22222
33333-33333-33333-33333
44444-44444-44444-44444
55555-55555-55555-55555
66666-66666-66666-66666
77777-77777-77777-77777
88888-88888-88888-88888
99999-99999-99999-99999
```

Dopo aver modificato il file `OUTPUT.TXT` chiudetelo.
Dopo questi passaggi, FAKE SERVER si metterà in modalità di ascolto sulla porta da voi scelta. In questo caso la porta `80`.
Durante il momento in cui FAKE SERVER è in attesa di qualche client, vi sarà ricordato di riaprire FAKE SERVER ad operazione conclusa, ed andare ad eseguire la funzione associata al tasto `2`. Di questa funzione ne parleremo fra poco.



Non appena qualche applicazione si collegherà a FAKE SERVER gli saranno trasmessi i nostri comandi.

Ovviamente per riuscire a crakkare qualche software, potremmo essere costretti a fare diverse prove e quindi a modificare il file `OUTPUT.TXT` più volte.
Il file `OUTPUT.TXT` potrà essere anche aggiornato in tempo reale, ma fino a quando il software che stiamo cercando di crakkare non farà nuovamente richiesta di ricezione dei dati, le modifiche applicate non avranno effetto.
Se necessario chiudete e riaprite il software che state cercando di crakkare e cercate di fargli avviare un'altra richiesta al server di riferimento.
FAKE SERVER crea anche un file di LOG, che potrebbe tornarvi molto utile per vedere quali sono i comandi inviati da parte del software che stiamo cercando di crakkare. Il file in questione si chiama `LOG.TXT`
Di seguito un esempio del LOG in cui sono stati salvati i comandi inviati da parte di Application Challenge 2 ed i comandi inviati da Google Chrome, durante dei tentativi di connessione manuale verso 127.0.0.1 che ho eseguito per fare delle prove.

```
Il Favoloso Mondo di Leo
https://ilfavolosomondodileo.wordpress.com/

FAKE SERVER 1.0 - LOG

-----------------------------------------------------------------------------------------------
19/04/2023  3:18:33 hackthissite.org 80
-----------------------------------------------------------------------------------------------

GET /app2.php?pass=kB2F.b0-sJS,k HTTP/1.0
Host: appchall2.hts

-----------------------------------------------------------------------------------------------
19/04/2023  3:18:35 hackthissite.org 80
-----------------------------------------------------------------------------------------------

GET /app2.php?pass=kB2F.b0-sJS,k HTTP/1.0
Host: appchall2.hts

-----------------------------------------------------------------------------------------------
19/04/2023  3:18:44 hackthissite.org 80
-----------------------------------------------------------------------------------------------

GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="99", "Google Chrome";v="109", "Chromium";v="109"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/109.0.0.0 Safari/537.36
Accept:
```

```
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: it,en;q=0.9
Cookie: HFS_SID_=0.213758566882461
dnt: 1
sec-gpc: 1

-----------------------------------------------------------------------------------------------
19/04/2023  3:18:46 hackthissite.org 80
-----------------------------------------------------------------------------------------------

GET / HTTP/1.1
Host: 127.0.0.1
Accept-Encoding: identity
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/90.0.4430.70 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Sec-Fetch-Mode: navigate
Connection: close

-----------------------------------------------------------------------------------------------
19/04/2023  3:19:03 hackthissite.org 80
-----------------------------------------------------------------------------------------------

GET /app2.php?pass=kB2F.b0-sJS,k HTTP/1.0
Host: appchall2.hts
```

Per terminare FAKE SERVER vi basterà semplicemente chiudere la finestra. Se volete provare a crakkare Application Challenge 2 con FAKE SERVER facendo uso del file `OUTPUT.TXT` integrato, potete utilizzare i seguenti codici product key.

```
11111-11111-11111-11111
22222-22222-22222-22222
33333-33333-33333-33333
44444-44444-44444-44444
55555-55555-55555-55555
66666-66666-66666-66666
77777-77777-77777-77777
88888-88888-88888-88888
99999-99999-99999-99999
```

Potete modificare il file `OUTPUT.TXT` come meglio credete e testare nuovi product key. La cosa importante è che rispettiate la sintassi originale dei comandi che dovranno essere dati in pasto ad Application Challenge 2. Per evitare problemi, vi indirizzo sempre all'articolo SOLUZIONE: Hacking This Site – Application – Application Challenge 2 dove potrete vedere i comandi che in condizioni normali vengono ricevuti da Application Challenge 2. Sulla base di questi comandi potrete modificare il file `OUTPUT.TXT`

Adesso dovrebbe esservi più chiaro perchè ogni tanto, può capitare di scaricare CRACK che ci invitano a modificare il file hosts o a disconnetterci provvisoriamente da internet.

Sicuramente CRAKKARE Application Challenge 2 non ci renderà ricchi in termini economici ma ci insegnerà molto.

Vi ricordate che FAKE SERVER proponeva di riaprire il file `SERVER.BAT` ad operazione conclusa ed eseguire la funzione associata al tasto `2`?

Questa funzione permette di ripristinare lo stato iniziale del file hosts ed annullare quindi il dirottamento, ed è importante eseguirla ogni volta che finite le vostre operazioni con il server.

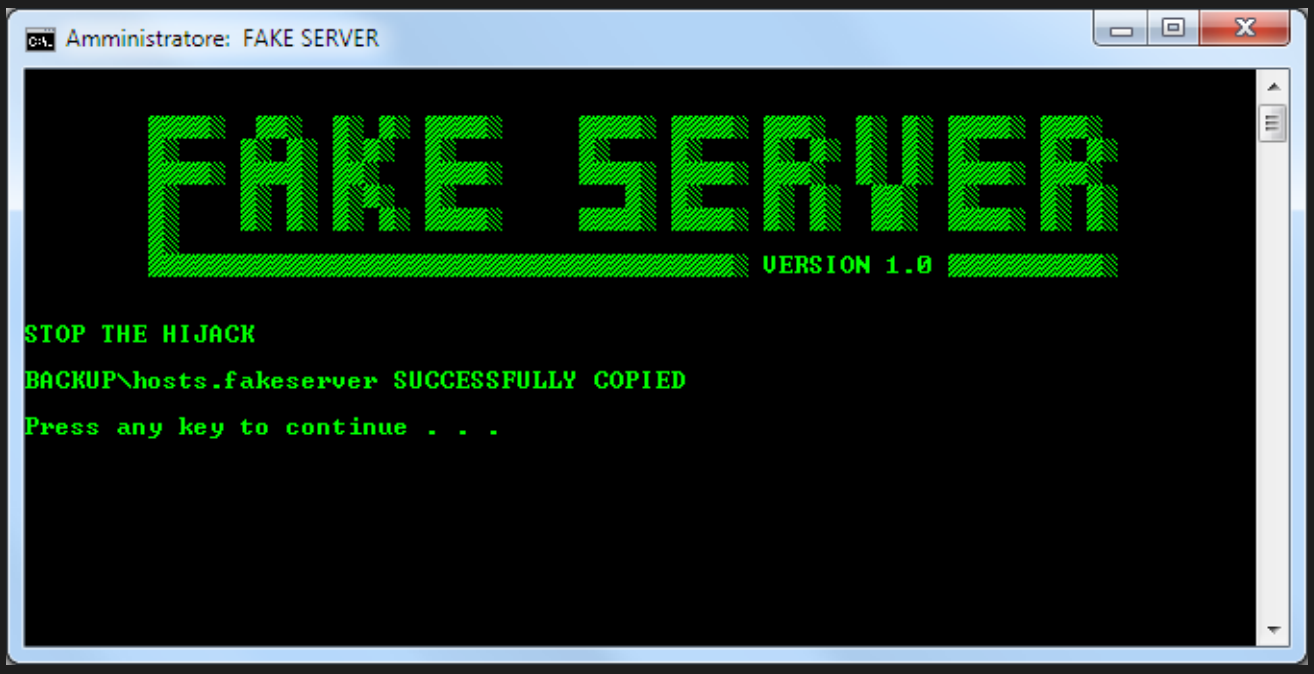
Direi che è fondamentale capire il comportamento di questa funzione.

FAKE SERVER in seguito alla vostra immissione di hostname e numero di porta, salverà un record in coda al file hosts ma non prima di aver creato due copie di backup.

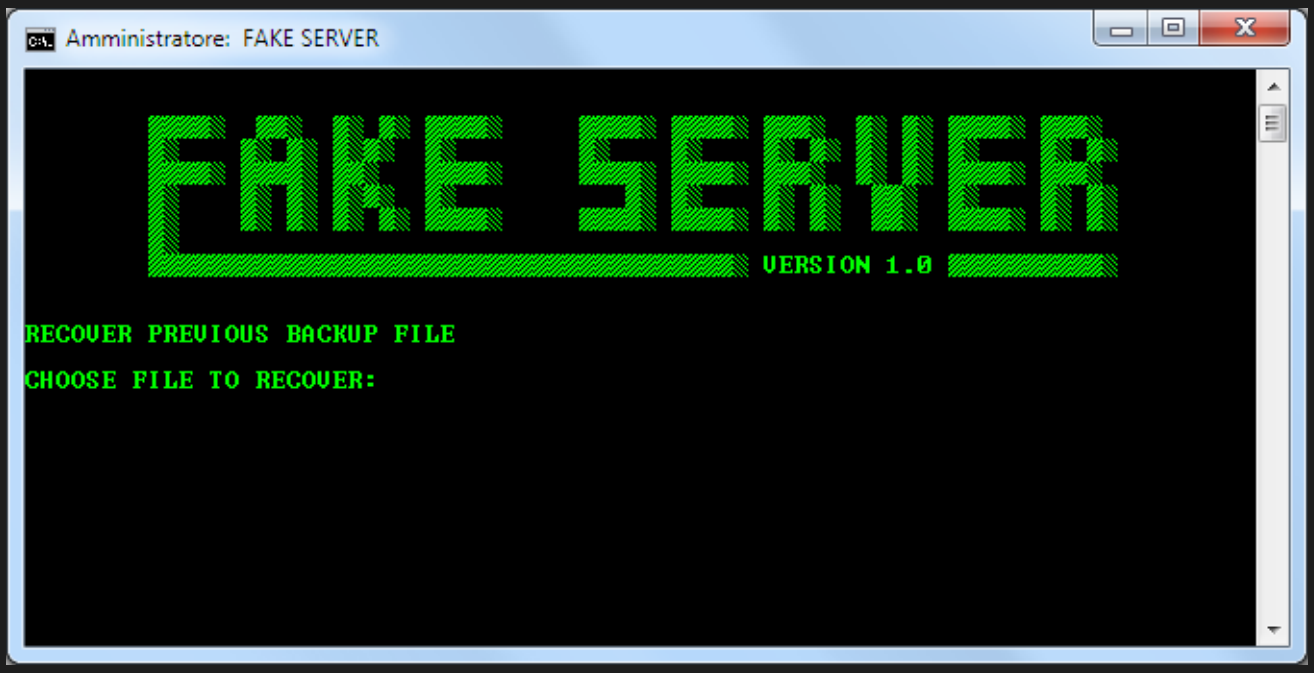
```
BACKUP\hosts.fakeserver
BACKUP\hosts.20230420151207
```

In questo esempio troviamo un file che si chiama `hosts.fakeserver` legato alla sessione attuale. Troviamo inoltre il file `hosts.20230420151207` che è anch'esso legato alla sessione attuale, ma a meno che non decidiate di avviare la funzione associata al tasto `4` che permette di eliminare qualsiasi BACKUP del file hosts creato in precedenza, questa copia rimarrà sempre conservata in modo tale che la possiate ripristinare in caso di problemi.

L'estensione di questo file vi permette di capire quando è stato eseguito il backup del file hosts.
In questo esempio il backup è stato eseguito il `20/04/2023` alle ore `15:12:07`
Di seguito un'immagine che mostra il termine del dirottamento andando ad avviare la funzione associata al tasto `2`.

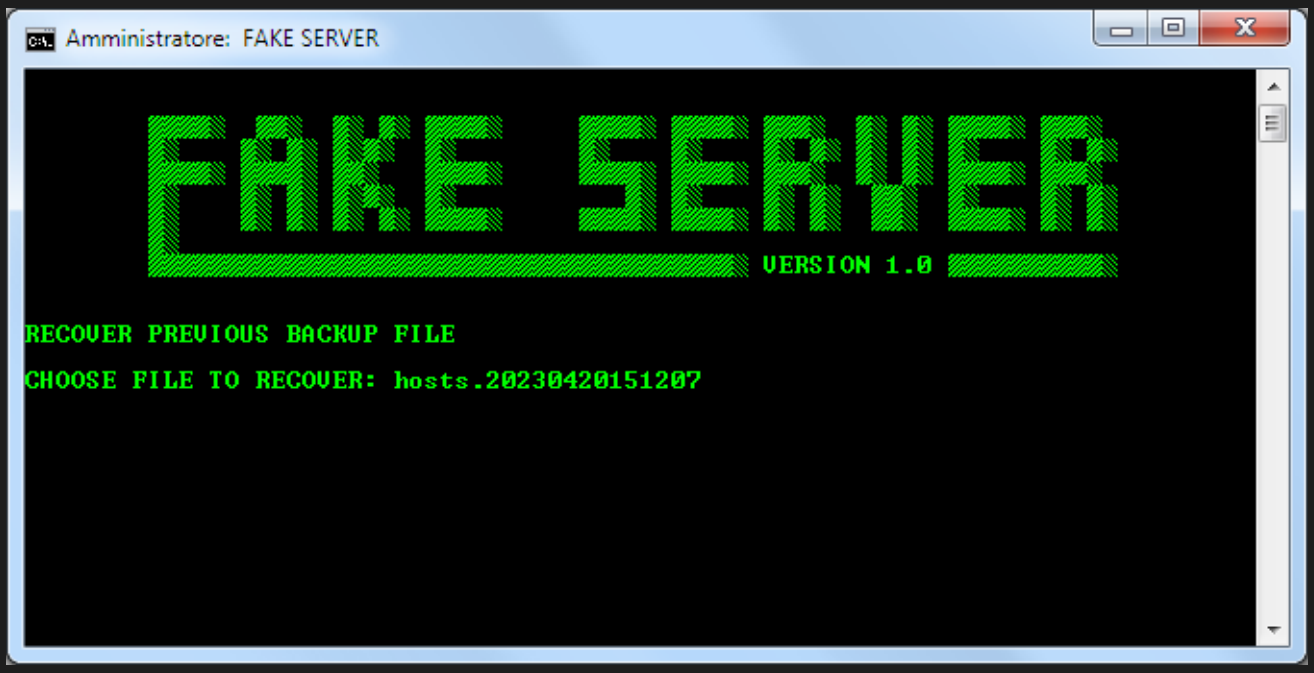


Per concludere, qualora ci dovessero essere problemi con le varie modifiche del file hosts, potete avviare la funzione associata al tasto `3`, che vi permette di selezionare manualmente un vecchio backup del file hosts e ripristinare quest'ultimo ad una condizione precedente.

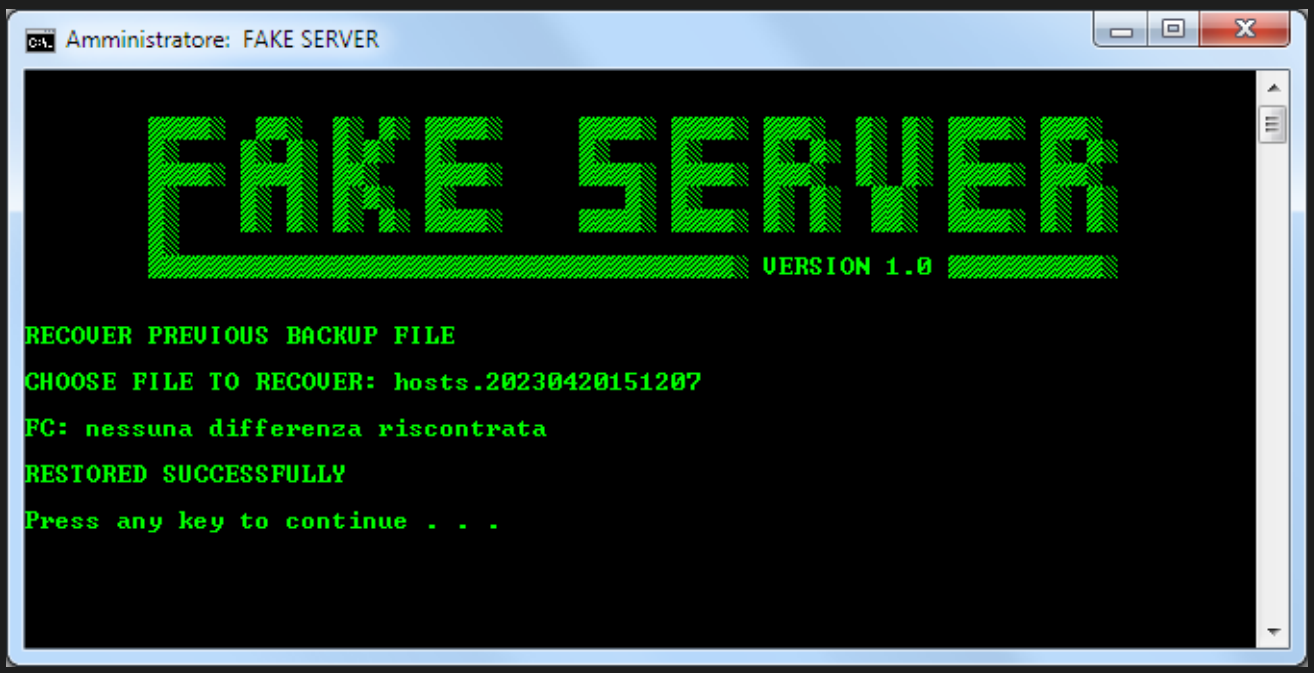


Piuttosto che digitare manualmente il nome del file da ripristinare, vi suggerisco di sfogliare i vari backup agendo sul tasto `TAB` della tastiera, fino a

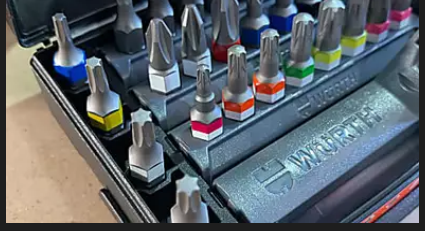
quando non comparirà la voce relativa al file di backup che volete ripristinare.



In seguito al ripristino del file di backup verrà inoltre eseguito un confronto binario fra lo stesso, ed il file hosts attualmente in esecuzione nel computer, per assicurarsi che il ripristino sia stato eseguito con successo.



Avete provato a usare FAKE SERVER?
Conoscete un programma simile?
Siete riusciti a crakkare qualche programma usando questo sistema?
Scrivete eventuali BUG o suggerimenti per migliorare FAKE SERVER e raccontatemi le vostre esperienze nei commenti.



aprile 20, 2023 Admin SOLUZIONI CRACKING

BATCH, CLIENT, CRACKING, DOWNLOAD, FAKE SERVER, NETCAT, PACKET SNIFFER, PRODUCT KEY, PROGRAMMA, SHELL, SNIFFING, SOFTWARE

Rispondi

Scrivi qui il tuo commento...