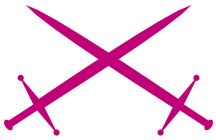

CyberForensics: Analisi di un sito Web

Stefano Maccaglia

Grado di difficoltà

Una delle più affascinanti aree dell'attività di Sicurezza è quella legata alla Forensics. Solitamente questo termine viene sempre collegato alla serie televisiva CSI, ormai divenuta parte integrante dell'immaginario condiviso di molte centinaia di migliaia utenti televisivi in giro per il mondo (Grissom rulez!).

In CSI, come poi accade nella realtà delle indagini di Polizia Scientifica, grazie ad una commistione inestricabile di capacità analitiche, tecnologie e senso pratico, si cerca di ricostruire la scena del Crimine per comprenderne le dinamiche e le condizioni di contorno. Questo lavoro, laddove possibile, aiuta molto gli investigatori nel dipanarsi delle indagini.

Un lavoro del genere è utile anche in situazioni di incidente informatico che vedono vittima dei servizi pubblicati via internet come ad esempio pagine web o ftp. In effetti ci sono dei paralleli possibili tra i metodi scientifici adottati dalle forze dell'ordine sulla scena di un crimine e quelli adottati dai tecnici informatici che analizzano un sito web, una directory o un ambiente logico. È obiettivo di questo articolo illustrare queste metodologie e mostrarne la scientificità offrendo congiuntamente una panoramica sugli strumenti a disposizione di un tecnico che debba analizzare un ambiente Web.

Chi scrive ritiene che gli elementi chiave di un metodo scientifico siano l'osservazione sperimentale di un evento (naturale o sociale), la formulazione di una ipotesi generale sotto cui questo evento si verifichi, e la possibilità di verifica dell'ipotesi mediante osservazioni successive.

Uno degli elementi essenziali affinché un complesso (limitato o meno) di conoscenze possa essere ritenuto scientifico è la sua possibilità di essere falsificabile attraverso un'opportuna procedura, nel nostro caso la replicabilità dell'analisi attraverso l'uso di strumenti Open Source, in linea con quanto sostiene l'approccio del DFRWS, l'organismo internazionale più accreditato in materia di Forensics.

Dall'articolo imparerai...

- L'obiettivo di questo articolo è quello di introdurre alle tecniche di *Computer Forensics* usate nell'ambito web. Le tecniche adottate sono solo alcune di quelle di più di facile adozione e non necessitano di particolari accorgimenti. I software suggeriti sono stati scelti dall'autore sulla base di due principali fattori: la facile reperibilità e la possibilità di utilizzarli con licenza Open Source o Shareware.

Cosa dovresti sapere...

- Per una migliore comprensione dell'articolo sono consigliabili conoscenze generali di architetture client-server, in particolare legate al protocollo Http al codice HTML.

Questo approccio lo ritroviamo, peraltro, in altri ambiti d'azione, come per esempio nella Risk Analysis laddove le metodologie OSSTMM propugnate da Pete Herzog e dall'Istituto ISECOM ribadiscono al pari la necessità di usare software Open Source per poter rendere falsificabile e pubblicamente ripetibile ogni esperienza di analisi da parte di un Pen Tester.

Rimane da sottolineare che quanto esposto in questo articolo vuole essere introduttivo ad una scienza, la *Cyberforensics* appunto, che è ancora in fase di *costruzione* e che è promanazione dalla scienza forense più tradizionale, ma dalla quale si differenzia per degli ovvi e necessari distinguo sia nelle tecniche che nelle modalità operative.

Rimangono validi degli assunti fondamentali in merito alle evidenze raccolte nell'indagine e cioè:

- Ammissibilità, per poter essere utilizzate in sede legale,
- Autenticità, ovvero l'essere strettamente correlate ai media posti sotto analisi,

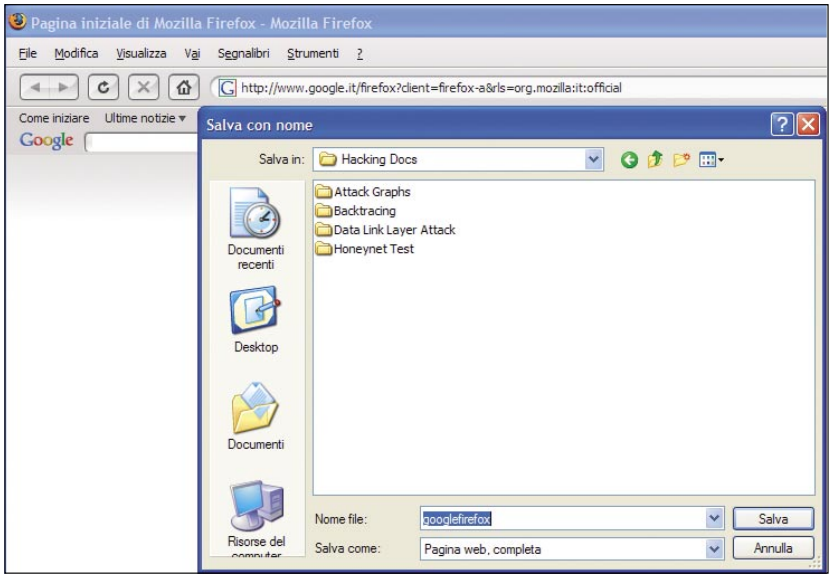

Fig. 1: *Salvataggio Pagina Web Completa – Mozilla Firefox*

- Completezza, in quanto le prove devono poter coprire tutto quello che si definisce estraibile dai sistemi di memorizzazione analizzati,
- Affidabilità, per non sollevare dubbi sulla loro autenticità e sulla procedura utilizzata per la loro rilevazione,
- Credibilità, permettendo a chiunque di ricostruire il processo che ha portato alla rilevazione delle evidenze ottenendo gli stessi risultati.

Se questi dettami sono però *piuttosto semplici* da applicare in ambiti quali il recupero di dati o la crittanalisi (pure aree essenziali della *Cyberforensics*)

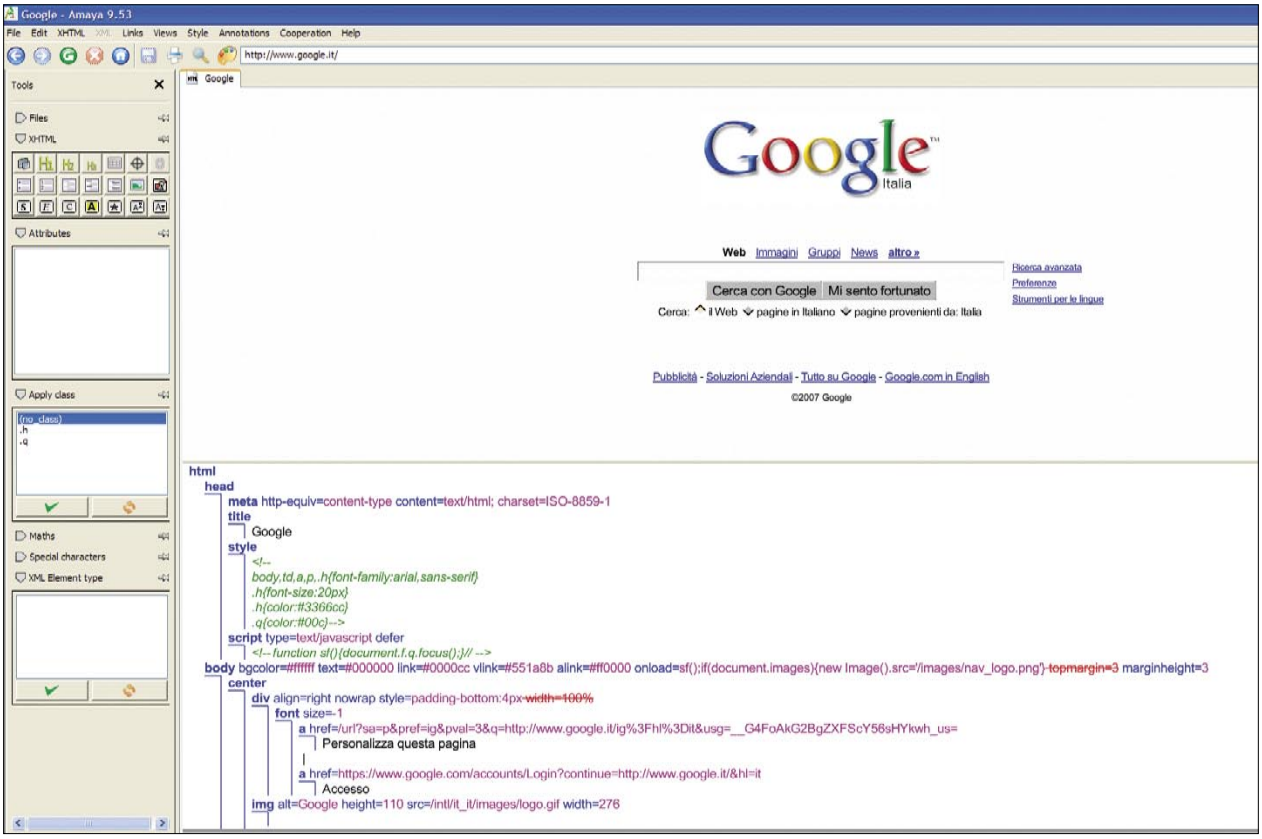

Fig. 2: *Amaya Browser/Editor*

laddove i dati sono il più delle volte a disposizione dell'investigatore e solo di questo per tutta la durata delle indagini, in ambito web si deve procedere attraverso una rilevazione spesso rapida e mirata delle prove e delle infrazioni in quanto i siti possono venire chiusi, alterati o spostati su altre strutture inficiando il lavoro fino a quel momento svolto. Ciò non deve far presupporre che si possa così andare contro i principi sopra esposti, ma è chiaro che soprattutto in merito alla completezza e all'ammissibilità si entra in ambiti non più solo tecnici. Per non parlare dei casi in cui il sito truffaldino su cui si investiga venga pubblicato in una nazione che non ha una stringente regolamentazione del web e dei suoi contenuti...

Forensics di un sito web

Praticamente ogni giorno chi lavora nella Sicurezza informatica viene a contatto con notizie di crimini perpetrati ai danni di aziende più o meno note da fantomatici malfattori che sfruttano vulnerabilità note o meno note per rubare credenziali di accesso, alterare l'aspetto del sito web, o peggio trafugare chissà quali segreti aziendali. In realtà, a causa di ben note problematiche quali la mancanza di manutenzione, di aggiornamenti e di controlli, molto spesso un'attività di defacement o di furto di identità via Internet avviene in maniera molto semplice, troppo semplice... Chi scrive dedicherà spazio in futuro all'argomento, resta da considerare che fino a poco tempo fa si poteva addirittura svolgere un *mass-defacement* appoggiandosi ai più noti motori di ricerca. Negli ultimi tempi molti passi avanti sono stati fatti nella cultura della Sicurezza e quindi anche il più pigro dei sistemisti è consapevole che gli aggiornamenti critici devono essere fatti su base settimanale e non una volta l'anno. Ma rimangono ancora molto ampi, in alcuni casi terribilmente ampi, i tempi di uscita di una patch per una vulnerabilità applicativa critica da parte delle grandi aziende.

Questo è evidente soprattutto per servizi comuni quali la posta elettronica e il web.

In questi casi, congiuntamente con l'attività di *Incident Response*, è importante svolgere una corretta procedura di polizia scientifica dell'ambiente applicativo attaccato.

Identificare le aree compromesse, isolarne i confini e delimitare i flussi comunicativi somiglia molto alle operazioni di routine che vediamo nelle puntate di CSI. C'è inoltre da raccogliere il DNA...

Nel nostro caso i log del sistema corrotto, del router, del firewall degli

Listato 1: *Frammento codice HTML di Google salvato localmente:*

```
</label></font></td></tr></tbody></table></form><br><br><font size="-
                    1"><a href="http://www.google.it/intl/it/ads/
                    ">Pubblicità</a> - <a href="http://www.google.it/
                    services/">Soluzioni Aziendali</a> - <a href="http:
                    //www.google.it/intl/it/about.html">Tutto su Google</
                    a> - <a href="http://www.google.com/ncr">Google.com in
                    English</a></font><p><font size="-2">©2007 Google</
                    font></p></center></body></html>
```

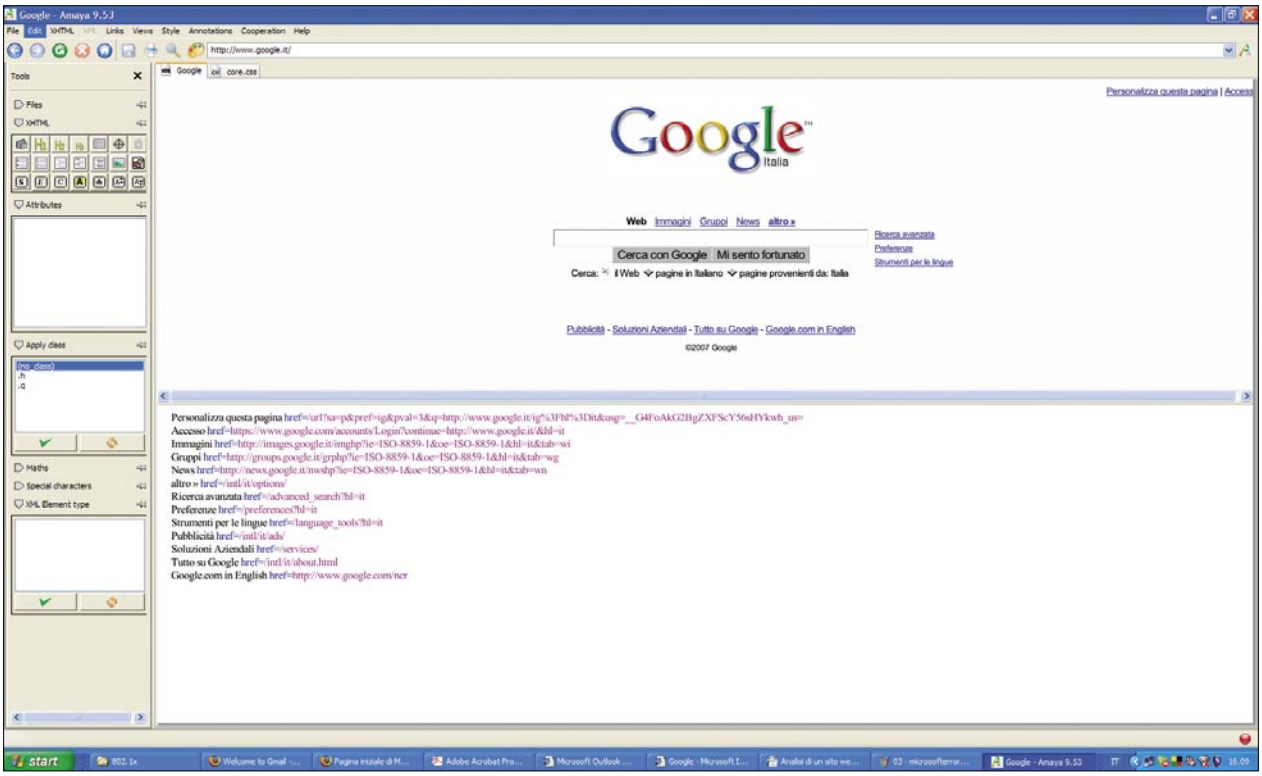

Fig. 4: *Amaya vista Collegamenti (Links)*

apparati IDS/IPS, tutto questo è utile a comprendere chi o come si è potuto perpetrare l'attacco.

Il nostro frammento di Dna è un indirizzo IP magari nascosto dentro una catena di proxy che ci rendono difficile identificare quello originario, quello da cui il nostro malfattore ha lanciato l'attacco.

Ma occorre anche capire i modi di accesso: quale vulnerabilità ha sfruttato, quali alterazioni ha potuto svolgere nelle macchine vittima, dobbiamo pensare come colui che ha attaccato, vedere tutte le possibili aperture che il codice o il server gli hanno fornito.

Analizzare minuziosamente la struttura di un sito web e del server su cui poggia diviene quindi un compito fondamentale nell'attività di indagine per crimini commessi attraverso questo percorso.

Questa attività comprende un controllo iniziale sul codice sorgente delle pagine web e successivamente dopo una attività di archiviazione del web-site, un controllo sugli script e sui processi *server-side* con l'obiettivo di comprenderne l'architettura applicativa e le caratteristiche, nonché tutte le possibili alterazioni.

Listato 2: *frammento codice HTML di Google:*

```
</label></font></td></tr></table></form><br><br><font size=-1><a href=/intl/
it/ads/>Pubblicità</a> - <a href=/services/>Soluzioni Aziendali</a> - <a
href=/intl/it/about.html>Tutto su Google</a> - <a href=http://www.google.com/
ncr>Google.com in English</a></font><p><font size=-2>&copy;2007 Google</
font></p></center></body></html>
```

Sulla stessa linea è il comportamento in presenza di phishing o pharming. Dobbiamo analizzare il codice delle pagine con cui si produce il tentativo di frode, capire i meccanismi di replica del sito vittima, valutare le differenze tra il sito vittima e la sua copia, individuare verso quale destinazione vanno le credenziali dell'utente una volta che questo cade vittima del raggiro.

Schematizzando i passi che compongono il lavoro di analisi sono i seguenti:

- Analisi preventiva via Browser,
- Analisi sincronica e diacronica del codice HTML,
- Schematizzazione della struttura del sito e dei collegamenti (*Link*) interni ed esterni,
- Comparazione delle versioni del sito e del suo codice,
- Analisi degli header HTTP e dei processi lato Server (*Server-side*),
- Analisi delle informazioni di sessione e di accesso,
- Report dell'attività di analisi.

Di tutti questi passi l'unico che il presente articolo non toccherà sarà proprio il Report, attività legata più ai template e alle best practices che variano da azienda ad azienda e da un professionista all'altro e per la quale occorrerebbe un articolo a parte, vista la quantità di differenti formati usati comunemente.

Analisi preventiva

Come in tutte le ricerche scientifiche basate su prove empiriche, il nostro punto di partenza è la cosiddetta analisi preventiva. È opportuno visualizzare le singole pagine web costituenti un sito navigandone le varie

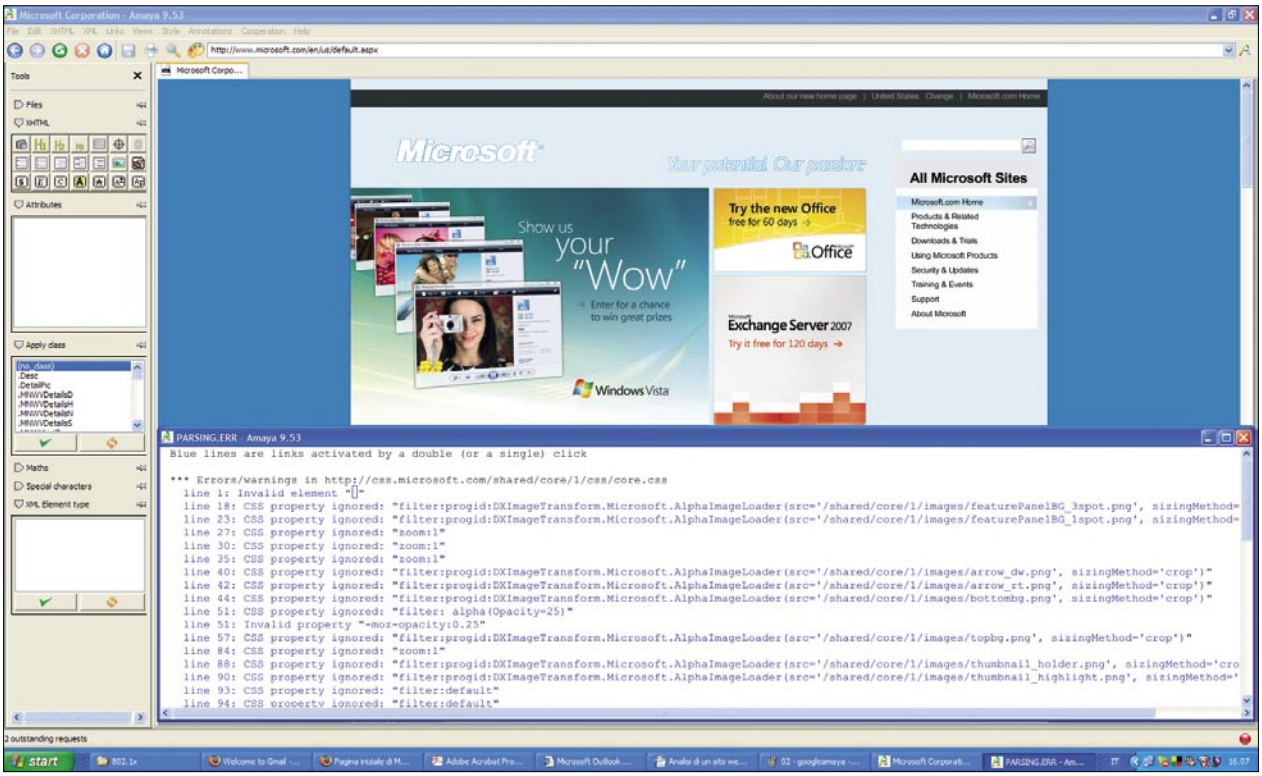

Fig. 3: *Vista del parser Amaya sul sito www.microsoft.com*

parti come farebbe un qualsiasi utente e raccogliendo, laddove si ritenga interessante, il codice HTML delle varie pagine a partire dall'index.

Il codice HTML sorgente di una pagina ci può offrire numerose informazioni circa il suo creatore, e i collegamenti contenuti al suo interno possono aiutarci a definire la mappa della struttura del sito stesso.

Questa analisi è ancora più importante quando ci troviamo ad esaminare il codice di un sito fantoccio usato per *Phishing* o *Pharming*. Per controllare il codice, in questa fase, è opportuno salvare localmente sul nostro PC ogni pagina.

Ci sono una serie di importanti aspetti di Forense collegati con questa semplice e apparentemente insignificante attività.

Anzitutto grazie ai collegamenti che individuiamo e alla ramificazione del sito nella rete possiamo comprendere le finalità del sito e la sua posizione rispetto al resto della *Big Internet*: indicatori quali il numero e il tipo di pagine esterne collegate con il sito, le modalità e la semantica con cui è indicizzata la pagina nei ranking di *Google*, *Altavista*, *Yahoo* e gli altri motori di ricerca sono tutti elementi probanti una popolarità del sito.

Altro elemento molto interessante è dato dall'individuazione dei componenti e dalla lettura del contenuto del sito: quale messaggio veicola, quali soggetti sono coinvolti, quali argomenti tratta.

Comprendere e catalogare il sito attraverso questa prima fase è un ottimo esercizio e ci permette di scegliere opportunamente gli strumenti dei passi successivi. Resta da evidenziare come, se volessimo già in questa fase raccogliere delle pagine in locale per analisi, si debbano risolvere alcuni aspetti organizzativi. Il primo è dato dal fatto che le pagine HTML di un sito moderno includono una più o meno vasta serie di altri file senza i quali la pagina stessa non avrebbe l'aspetto che ha. Le immagini sono il più ovvio esempio, ma si possono aggiungere *Formattazioni*, *Stili*, *Javascript* e altri tipi di file. In molti casi i link a questi file sono relativi e non assoluti, ciò significa che essi non verranno salvati se copiamo localmente la pagina, né verranno visualizzati se poi la apriamo sul nostro computer. Quindi se vogliamo assicurarci sulla completa visualizzazione della pagina originale si procederà attraverso uno dei seguenti modi: aggiornando i link relativi di questi file nel codice della pagina copiata oppure salvando localmente nello stesso repository del file HTML i file a corredo. In realtà alcuni browser recenti, come *Mozilla Firefox* o *Opera*, rispondono a questo problema salvando ogni file associato quando una pagina web è salvata in modalità completa come in Fig. 1.

Questi file saranno salvati in una directory che verrà creata nella

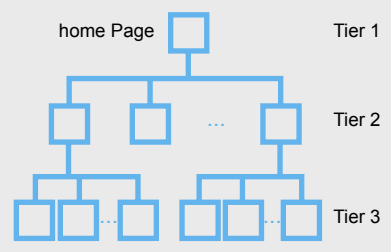

Fig. 6: *Esempio di Struttura Web*

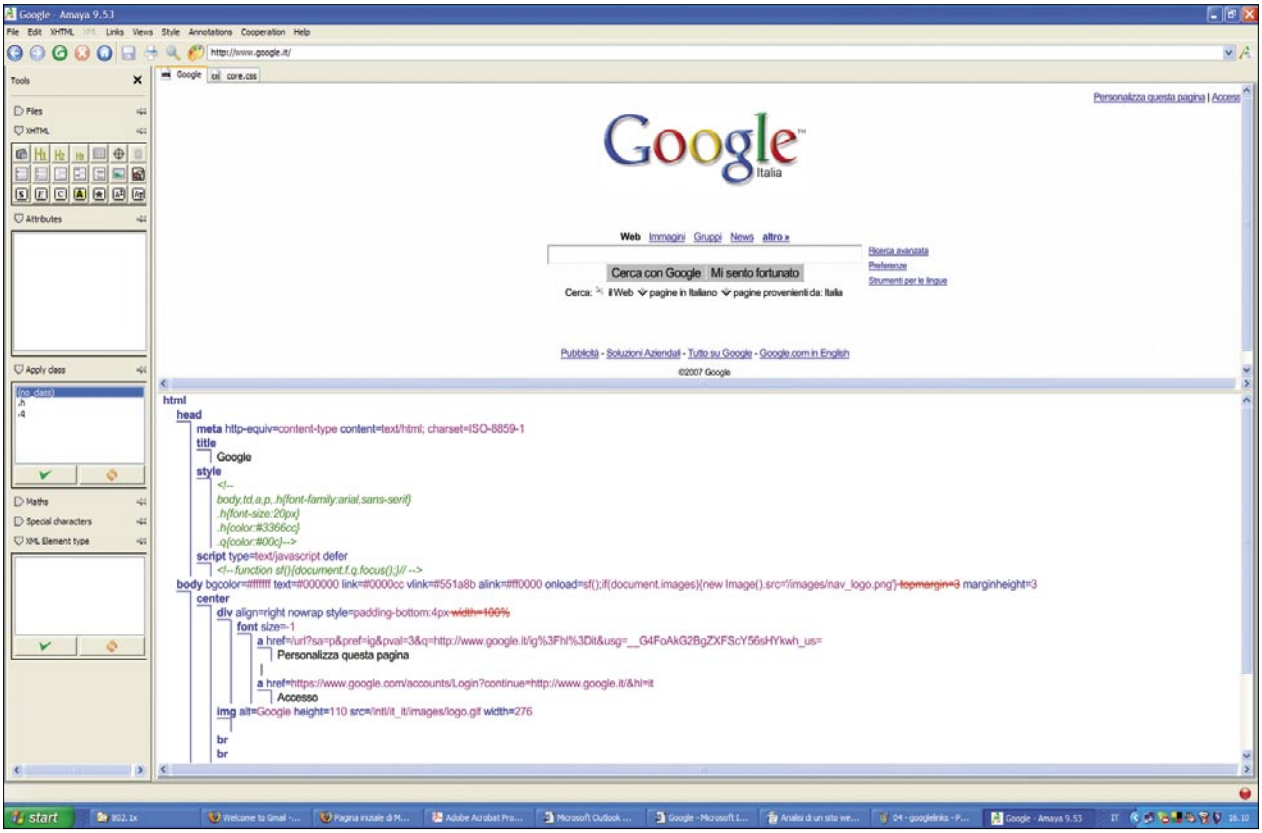

Fig. 5: *Amaya -> Vista Struttura*

Terminologia

- *Defacement – Defacing* (termine inglese che, come il suo sinonimo defacement, ha il significato letterale di sfregiare, deturpare, in italiano reso raramente con defacciare) nell'ambito della sicurezza informatica ha il significato di cambiare illecitamente la home page di un sito web (la sua faccia) o modificarne, sostituendole, una o più pagine interne.
- *DFRWS* = Digital Forensic Research Workshop.
- *HTML* – (acronimo per *Hyper Text Mark-Up Language*) è un linguaggio usato per descrivere i documenti ipertestuali disponibili nel Web. Non è un linguaggio di programmazione, ma un linguaggio di markup, ossia descrive il contenuto, testuale e non, di una pagina web. Punto HTML (*.html*) o punto HTM (*.htm*) è anche l'estensione comune dei documenti HTML.
- *HTTP* – è l'acronimo di *Hyper Text Transfer Protocol* (protocollo di trasferimento di un ipertesto). Usato come principale sistema per la trasmissione di informazioni sul web. Le specifiche del protocollo sono attualmente in carica al W3C (*World Wide Web Consortium*).
- *HTTPS* – Con il termine HTTPS ci si riferisce al protocollo HTTP (*Hyper Text Transfer Protocol*) utilizzato in combinazione con lo strato SSL (*Secure Socket Layer*); la porta standard dedicata a questo servizio è la 443/TCP. In pratica viene creato un canale di comunicazione criptato tra il client e il server attraverso lo scambio di certificati; una volta stabilito questo canale al suo interno viene utilizzato il protocollo HTTP per la comunicazione.
- *OSSTMM – Open Source Security Testing Methodology Manual* (Manuale metodologico per l'analisi della Sicurezza Open Source). Nato dal lavoro di Pete Herzog e del team chiamato Idealhamster alla fine del 2000, questo approccio fornisce una metodologia per svolgere i test di sicurezza. Un test di sicurezza sotto questa chiave di lettura diviene una misura accurata della sicurezza operativa applicata in azienda, misura basata su fatti comprovabili e ripetibili e non su semplici assunzioni o ipotesi. Questa metodologia è *Open* in virtù del fatto che è libera da vincoli corporativi e politici. È poi Open Source in quanto permette un dibattito libero e svincolato dagli ostacoli della proprietà intellettuale sulle sue procedure operative da parte dei suoi sostenitori e delle persone che con essa lavorano, garantendo uno sviluppo democratico della stessa metodica e delle sue appendici operative.
- *Pharming* – tecnica di cracking, utilizzata per ottenere l'accesso ad informazioni personali e riservate, con varie finalità. Grazie a questa tecnica, l'utente è ingannato e portato a rivelare inconsapevolmente a sconosciuti i propri dati sensibili, come numero di conto corrente, nome utente, password, numero di carta di credito ecc. A differenza del Phishing questa tipologia di truffa è tecnicamente più complessa dovendo operare alterazioni nella macchina della vittima o nel sistema di comunicazione (il DNS, per esempio).
- *Phishing* – attività truffaldina che sfrutta una tecnica di ingegneria sociale, ed è utilizzata per ottenere l'accesso a informazioni personali o riservate con la finalità del furto di identità mediante l'utilizzo delle comunicazioni elettroniche, soprattutto messaggi di posta elettronica fasulli o messaggi istantanei, ma anche contatti telefonici.
- *Spider* o *Crawler* – Un crawler (detto anche spider o robot), è un software che analizza i contenuti di una rete (o di un database) in un modo metodico e automatizzato, in genere per conto di un motore di ricerca. Un crawler è un tipo di bot (programma o script che automatizza delle operazioni). I crawler solitamente acquisiscono una copia testuale di tutti i documenti visitati e le inseriscono in un indice.
- *SSL – Secure Sockets Layer* (SSL) è un protocollo progettato dalla *Netscape Communications Corporation*, autrice del famoso browser *Netscape Navigator* per realizzare comunicazioni cifrate su Internet. La versione 3.0, rilasciata nel 1996, è stata utilizzata come base di sviluppo per il protocollo *Transport Layer Security* (TLS). TLS è un protocollo standard IETF che è definito nella RFC 2246 (*The TLS Protocol Version 1.0*). Questi protocolli utilizzano la crittografia per fornire sicurezza nelle comunicazioni su Internet e consentono alle applicazioni client/server di comunicare in modo tale da prevenire il *tampering* (manomissione) dei dati, la falsificazione e l'intercettazione.

stessa locazione in cui verrà salvata la pagina web. Questo significa che se io salvo la pagina *googlefirefox.html* in una directory come pagina completa, allora mi ritroverò anche una sottodirectory definita *googlefirefox_files* che conterrà tutte le immagini, gli stili, e i file riferiti al file originale. Inoltre la maggior parte dei link di questi file saranno stati aggiornati in modo da puntare logicamente alla locazione in cui sono presenti le copie locali. Uso il termine maggior parte non a caso, perché *Firefox* non è in grado di aggiornare i link che sono inclusi come parametri di funzioni *JavaScript* quali i rollover o i *pop-up*. A parte queste eccezioni la pagina salvata e tutti i file ad essa legati potranno essere aperti da un browser (non necessariamente lo stesso *Mozilla*) e la pagina apparirà pressoché identica all'originale. Le differenze più importanti, oltre al fatto degli script e dei *pop-up*, saranno legate alle modifiche che *Firefox* svolge nel codice HTML che salva. Quindi le differenze saranno più visibili nel codice che nell'aspetto del sito web. Il perché di queste modifiche è da ricercare probabilmente nel tentativo, che il browser svolge, di salvare pagine in formato HTML valido evitando cioè i rischi di tag non complete o di attributi relativi privi di valori nelle pagine copiate. Ciò però porta le pagine a non poter essere comparate facilmente.

Un esempio è estratto dal codice HTML salvato dal file *googlefirefox.html*, nella porzione riferita agli script (Listato 1) in rapporto al codice HTML della pagina originale (vedi Listato 2).

Come si può notare *Firefox* ripropone gli stessi valori per gli attributi, ma nel file locale essi sono stati spostati in base all'ordine alfabetico delle loro tag. In aggiunta ha inserito delle nuove tag quali `<tbody><tr>` che alterano in maniera significativa la lettura comparativa del codice. Per evitare questa confusione è preferibile salvare le pagine in *Firefox* come Pagina Web, solo HTML o usare tool non interattivi per la copia come *Wget* o *HTTrack*. Per inciso, anche *Internet Explorer* o *Opera* o anche *Camino* e gli altri web browser di ultima generazione

Listato 3: *Frammento degli errori riportati dal sito www.microsoft.com*

```
Blue lines are links activated by a double (or a single) click
*** Errors/warnings in http://css.microsoft.com/shared/core/1/css/core.css
  line 1: Invalid element ""
  line 18: CSS property ignored: "filter:progid:DXImageTransform.Micros
                      oft.AlphaImageLoader(src='/shared/core/1/images/
                      featurePanelBG_3spot.png', sizingMethod='crop')"
  line 23: CSS property ignored: "filter:progid:DXImageTransform.Micros
                      oft.AlphaImageLoader(src='/shared/core/1/images/
                      featurePanelBG_1spot.png', sizingMethod='crop')"
  line 27: CSS property ignored: "zoom:1"
  line 30: CSS property ignored: "zoom:1"
  line 35: CSS property ignored: "zoom:1"
  line 40: CSS property ignored: "filter:progid:DXImageTransform.Microsoft
                      .AlphaImageLoader(src='/shared/core/1/images/arrow_
                      dw.png', sizingMethod='crop')"
  line 42: CSS property ignored: "filter:progid:DXImageTransform.Microsoft
                      .AlphaImageLoader(src='/shared/core/1/images/arrow_
                      rt.png', sizingMethod='crop')"
  line 44: CSS property ignored: "
```

Listato 4: *Risultato del diff tra un sito web reale ed una sua copia per phishing*

```
8c6
< <link rel="stylesheet" href="http://www.***.it/online/loginhome.fcc?[...]"
type="text/css" media="all" />
---
> <link rel="stylesheet" href="/lin/css/lin2obj.css"
type="text/css" media="all"/>
31c26
< <a href="login.htm?requester=signon" class="obibtn">Sign On</a>
---
> <a href=https://bancopostaonline.poste.it/bpol/[...]/Controller?requester=
signon class="obibtn">Sign On</a>
65c59
< <!-- COMMENT--></object>
[...]
< <IMG SRC="http://www.myspace.com/[...]/pers?s=1001068&t=11001461"
ALT=1 WIDTH=1 HEIGHT=1>
```

possono salvare tutti i file associati ad una pagina. L'alternativa pure percorribile è *NetCat*. Un'altra difficoltà che si incontra è dovuta al fatto che la maggior parte delle pagine web quando le scarichiamo in locale dal browser, non includono l'URL da cui provengono. Si dovrà salvare l'URL in un file separato o si dovrà inserire in un commento nella pagina web salvata manualmente e questo ci può rallentare nelle operazioni di analisi.

Possiamo comunque affermare che il supporto dei più comuni browser è sufficiente a svolgere la prima fase di analisi, ma se le cose si fanno più complesse si consiglia l'uso di un browser alternativo, molto utile già in questa fase di analisi: *Amaya*.

Amaya

Amaya è sviluppato dal *W3C Consortium* per tutti i più comuni sistemi operativi (Linux, Windows e Mac Os X) ed è attualmente giunto alla sua release 9.53. Come si può vedere dall'immagine seguente questo software semplifica enormemente le procedure di analisi permettendo in maniera diretta di poter controllare tag, stili e formattazione di un sito web, di raccogliere i dati dei link della pagina, di visualizzarne la struttura e i *cross-link* e di esportarne varie versioni potendo editarne il contenuto in modo completamente integrato.

Unico neo il problema di non poter lavorare con le pagine HTTPS in quanto non si riesce ad eseguire il GET della pagina, operazione con cui *Amaya* acquisisce e visualizza le pagine durante la navigazione.

Questo software è utilissimo nell'analisi comparativa di un sito in quanto prevede, tra le varie viste, quella sul codice HTML e quella sui collegamenti, nonché quella sugli errori come per esempio le tag non chiuse. Come per esempio questa pagina del sito Microsoft da cui ne abbiamo estratto solo una piccola parte.

Analisi Preventiva: locazione del Sito

Prima di scendere in dettaglio nell'analisi del codice HTML vorrei però indicare altri elementi dell'attività iniziale di indagine che sono essenziali per definire le caratteristiche del sito: la sua collocazione geografica e il *Provider*. Non scenderemo nel dettaglio, più che altro a causa dello spazio a disposizione, ma lasciamo ampi spazi all'approfondimento attraverso i link. L'indagine sulla collocazione geografica del sito partono dalla sua URL, è importante in questo senso interrogare anzitutto i server di registrazione del dominio per cercare di individuarne il proprietario.

Terminologia

- *W3C* – Nell'ottobre del 1994 Tim Berners Lee, considerato padre del Web, fondò al MIT (*Massachusset Instutute of Technlogy*), in collaborazione con il CERN (il laboratorio dal quale proveniva), un'associazione di nome *World Wide Web Consortium* (abbreviato W3C), con lo scopo di migliorare gli esistenti protocolli e linguaggi per il WWW e di aiutare il Web a sviluppare tutte le sue potenzialità.
- *Warez* – materiale, prevalentemente software, distribuito in violazione al copyright che lo ricopre. Storicamente il termine non è associato ad alcuno scopo di lucro, quanto piuttosto di profitto per il mancato acquisto del prodotto informatico: la vendita non è caratteristica del *Warez*. *Solitamente* ci si riferisce a distribuzioni di software ad opera di gruppi organizzati via Internet.

Congiuntamente con questa azione è opportuno *tracciare* il sito stesso sia via URL che via IP.

Queste operazioni possono essere condotte da riga di comando o via browser, per sintesi rimando all'esaustivo articolo presente su *Wikipedia* all'indirizzo: *http:///it.wikipedia.org/wiki/Whois*.

Per comprendere poi la sua collocazione occorre provvedere con tool quali visualroute o neotrace:

http://www.visualroute.com/ oppure *Neotrace Express* (per Windows), ormai integrato nella suite *McAfee*, ma ancora reperibile in rete come shareware.

Evidentemente molti degli spazi web che ospitano siti di phishing sono collocati in aree geografiche in cui la legislazione è piuttosto permissiva con gli abusi e i crimini informatici, ma spesso si trovano abusi provenienti anche da siti che offrono spazi gratuiti ed in questo senso una volta individuata la provenienza e le finalità illecite del sito stesso è opportuno segnalare all'abuse e al provider quanto scoperto.

Analizzare il sorgente HTML

Partiamo da un assunto dettato dall'esperienza: le pagine di un sito web commerciale sono sempre progettate e realizzate con l'obiettivo di renderle piacevoli e interessanti a livello estetico, ma raramente la pulizia del codice HTML è un obiettivo condiviso. Il risultato è che pagine complesse sono spesso rappresentate da codice HTML indecifrabile.

Per facilitare la lettura di questo codice *Firefox* fornisce, attraverso l'opzione già vista di View page Source, una colorazione diversa per ogni tag, ma se questo non è sufficiente (come sostiene chi scrive) è opportuno ricorrere all'ausilio di software quale HTML Tidy (la cui primaria funzione è correggere errori nel codice, ma che offre un sostanziale miglioramento della visualizzazione indentando le tag e cambiando l'aspetto stesso del codice… Il programma è liberamente scaricabile per tutte le maggiori piattaforme da *http://tidy.sourceforge.net/*), come suggeriscono molti miei amici sviluppatori, o ancora il nostro *Amaya*. In Tidy, attraverso il seguente comando:

```
# tidy -i originale.html >
modificato.html
```

Il file *originale.html* può essere formattato in maniera appropriata e ad una successiva analisi dell'output il file *modificato.html* mostrerà un codice più ordinato, sempre ché il codice stesso non presenti talmente tanti errori da risultare non processabile, come capita ad esempio se cerchiamo di svolgere questa operazione con il sito *www.microsoft.com* di cui sopra. Il codice HTML ci dice molte cose interessanti:

- La struttura del Sito,
- Chi e/o con cosa è stato creato,
- Gli errori.

Comprendere la struttura del Sito

Estrarre i riferimenti alle altre pagine, le immagini e gli script da una pagina web è un'operazione essenziale per

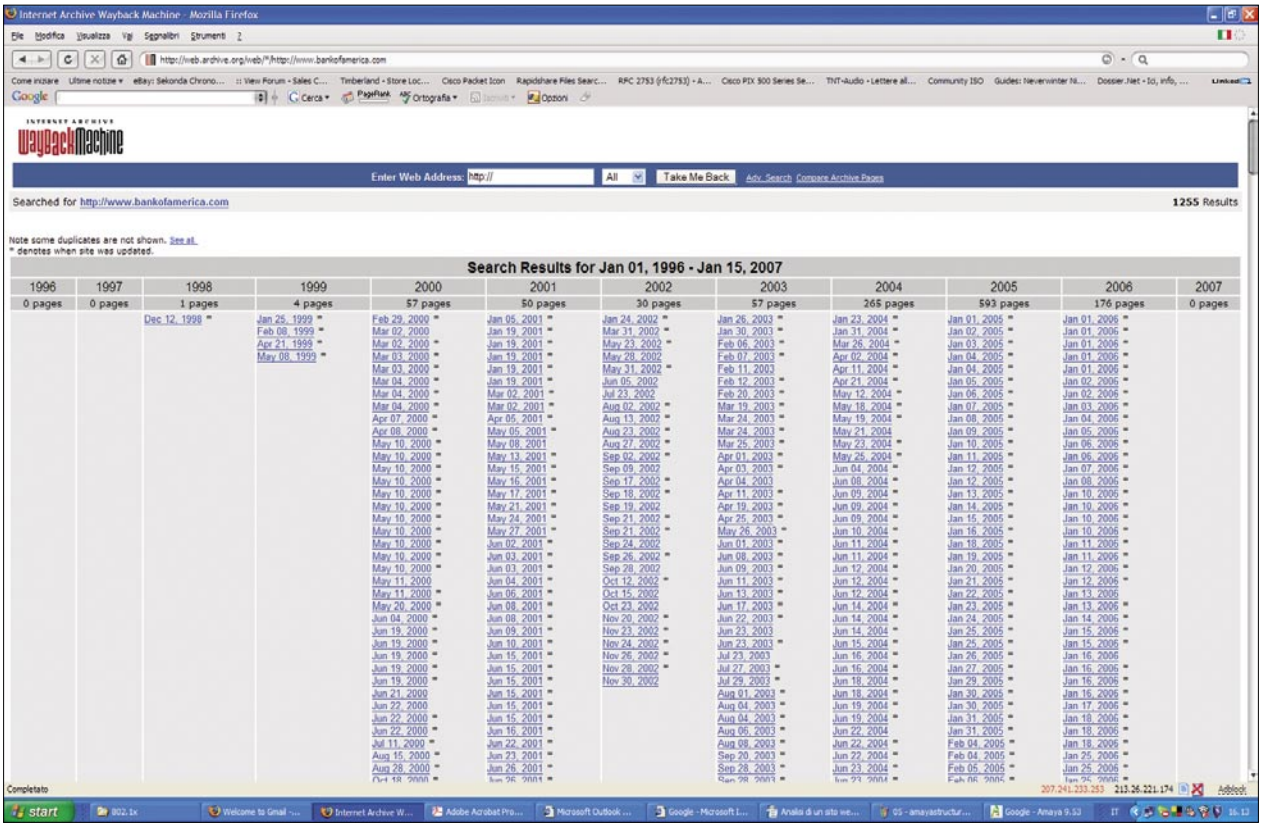

Fig. 7: *Wayback Machine*

Fig. 8: *Mail di Phishing con richiesta di aggiornamento delle Credenziali*

comprendere la struttura stessa del sito web, e a volte anche le finalità dello stesso, come nel caso di *Phishing*. Leggere nel codice HTML alla ricerca di collegamenti a volte diviene un lavoro improbo. Per questo chi scrive suggerisce di ricorrere ad uno script o piuttosto al già citato *Amaya* che ha tra le viste selezionabili quella relativa all'elenco di tutti i link presenti nella pagina, come si può vedere nella Figura 4.

Al termine di questa lettura del sito è raccomandabile schematizzare quanto individuato in modo da poter avere una mappa logica su cui svolgere le successive inferenze.

Schematizzazione della struttura di un sito web

Un'altra fase essenziale nell'analisi del Sito riguarda la sua struttura.

Amaya ci torna utilissimo anche in questo caso, infatti accedendo alla pagina e spuntando dal menù Views l'opzione *Show Structure*, o in alternativa con la combinazione di tasti `<CTRL> <o> <u>`, si ottiene una vista sulla struttura della pagina comprensiva di link, tag e formattazioni.

La visualizzazione della struttura è di capitale importanza per comprendere la meccanica del sito, la sua estensione logica, i suoi confini semantico/sintattici.

Schematizzare un sito web o anche solo una serie di pagine è un utile strumento nella valutazione della bontà del codice e dell'architettura, quindi anche della professionalità del webmaster. Da questa vista si possono comprendere le aree di azione nelle quali si muovono i contenuti veicolati dal sito e come sono correlati l'uno all'altro.

Il software con cui si è creato il sito

Un dato importante nell'analisi del codice è l'individuazione del software con il quale è stato creato. Molte pagine sono generate artigianalmente, o attraverso PHP o script *Perl*, ma molti altri sono sviluppati usando software quale *Microsoft FrontPage*, *Microsoft Word*, *Adobe GoLive*, o *Macromedia Dreamweaver*. Ogni software lascia dietro di se una firma in HTML quando genera il codice del Sito. Qualche volta questo codice prende la forma di uno style code specifico, altre volte un esplicito commento ci dà direttamente l'informazione. *Adobe GoLive* si identifica usando il meta tag dal nome generator:

```
<meta name="generator" content="Adobe
GoLive 4">
```

Macromedia Dreamweaver può essere identificato dal prefisso MM che usa con le funzioni JavaScript che spesso include nel codice HTML da lui prodotto:

```
function MM_preloadImages( ) { //v3.0
function MM_swapImgRestore( ) { //v3.0
function MM_findObj(n, d) { //v4.01
function MM_swapImage( ) { //v3.0
```

Microsoft FrontPage aggiunge un meta tag dal nome *ProgId*:

```
<META NAME="GENERATOR"
ONTENT="Microsoft FrontPage 5.0">
<META NAME="ProgId" CONTENT="FrontPag
e.Editor.Document">
```

Microsoft Word può generare pagine web convertendo documenti in

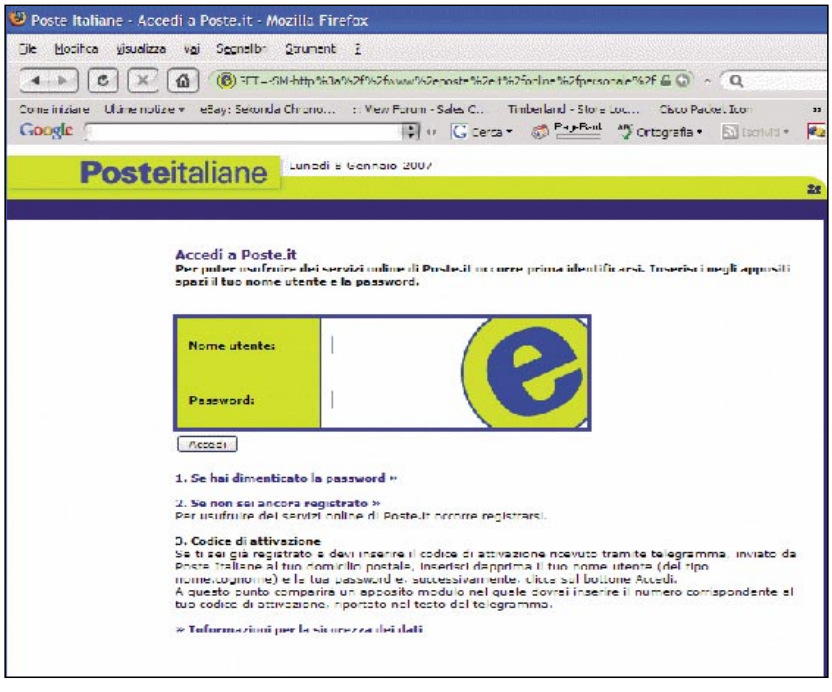

Fig. 9: *Pagina sorgente delle Poste Italiane*

HTML. Questo tipo di pubblicazione introduce meta tags del tipo:

```
<meta name=ProgId
content=Word.Document>
<meta name=Generator content="Microsoft
Word 10">
<meta name=Originator
content="Microsoft Word 10">
```

Anche quando queste tag sono state eliminate attraverso successive revisioni del codice, una pagina generata in Word può essere identificata dall'uso di stili che hanno il prefisso mso:

```
p.MsoNormal, li.MsoNormal,
div.MsoNormal
```

A volte si può anche incontrare, nell'analisi di codice HTML, una commistione di codice contenente varie signature e tag già indicate, tutto questo ci indica che il codice è stato alterato dalla sua forma originale, ma in alcuni casi è anche possibile individuare l'ordine con il quale si è proceduto alle varie revisioni passando per differenti software di editing.

Gli Errori

A chi possono interessare gli errori?

Gli errori ci dicono molte cose. Se essi sono macroscopici possono condurci anche molto lontano, in quelle regioni del sito in cui chi l'ha sviluppato non avrebbe mai voluto che arrivassimo...

Ma anche se sono di minore entità possono dirci molto, soprattutto nella profilazione di un criminale, che proprio attraverso il ripetere degli stessi errori nel codice, ci permette di collegare un sito ad un altro, un'attività ad un'altra, una sorta di biglietto da visita.

Questo argomento meriterebbe una trattazione a parte che non può essere svolta ora, posso però lasciare una serie di interessanti link all'argomento: *http://ftp.cerias.purdue.edu/pub/papers/steve-weeber/spaf-weeber-forensics.pdf*.

L'analisi degli errori nel codice per finalità di accesso, al pari come vedremo dell'analisi della configurazione del Server propedeutica all'individuazione di Porte e Servizi utilizzabili per un'intrusione, conducono alla Questione Etica...

Una qualsiasi attività che consista o che sia propedeutica alla penetrazione di Sistemi appartenenti a Terze Parti è da considerare un atto contrario all'etica, a meno che non si abbia un regolare permesso da parte delle autorità.

Può capitare però di imbatterci, nelle nostre navigazioni in un sito sviluppato per finalità illecite. In quest'ultimo caso raccomando ai lettori di mantenere un approccio etico, ma sottolineo anche che, laddove si è testimoni di reiterate attività criminali operate dallo stesso sito, è opportuno raccogliere ulteriori evidenze attuando azioni volte ad individuare dati più *nascosti*, come ad esempio i file nelle cartelle nascoste, in modo da poter avere un quadro più chiaro di chi abbiamo davanti, comprendere meglio i meccanismi della truffa e poter poi denunciare opportunamente l'abuso almeno davanti al provider e sostenere con le prove quanto denunciato. Il limite in questo caso è dato dalla volontà di raccolta documentale, è imperativo evitare l'uso di exploit aggressivi e tecniche di particolare virulenza che possano compromettere le funzionalità del sito e del server o l'inquinamento delle prove. È assolutamente da evitare qualsiasi attività che leda o alteri i dati e le funzionalità del Server, anche per il rischio che le finalità illecite vengano perpetrate attraverso una porzione nascosta di un sito legale e che usando tool aggressivi si finisca per andare contro la legge.

Analisi diacronica di un sito web – The Wayback Machine

Uno dei più importanti tool che ci viene in supporto nell'analisi dello storico di un sito web è fornito gratuitamente da *archive.org* con la ormai famosissima *Wayback Machine*. Questo servizio offerto dall'Internet Archive archivia pagine web sin dal 1996 indicizzando decine di migliaia di copie di siti web ogni giorno. Lo storico che la *Wayback Machine* ci offre di un qualsiasi sito web ha del miracoloso. Cercando per esempio il sito di una banca americana quale la *United Bank of America*, si trovano innumerevoli risultati a partire dal 1998. Come si può notare in Figura 7.

L'interazione con il tool è facilissima, al link: *www.archive.org/web/web.php* si inseriscono nel form in Figura i dati dell'URL da ricerca e cliccando su *Take Me Back* si otterranno tutti i risultati salvati dall'immenso sistema di archiviazione. Aprendo i vari

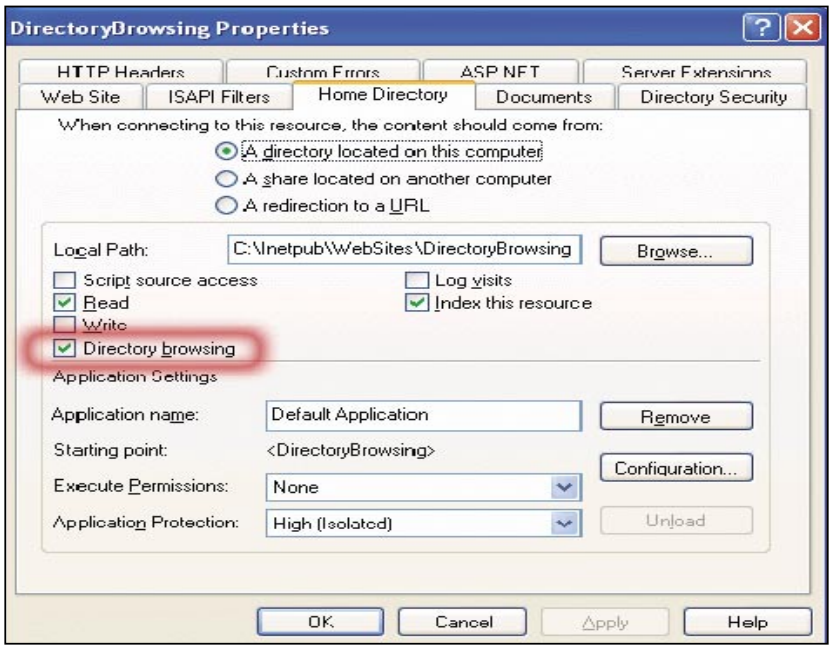

Fig. 10: *Directory Browsing*

link organizzati su base temporale si potrà visionare l'evoluzione del sito stesso. A volte attraverso i link assoluti di una vecchia versione di un sito web mi è capitato di trovarmi in pagine non più raggiungibili direttamente dal portale aziendale aggiornato.

Un altro grande vantaggio dell'uso della wayback machine è dato dalla possibilità di trovare risultati anche per siti non più attivi che possono essere associati a pratiche illegali quali il phishing appunto o il mass spamming.

Comparativa di pagine

Chi scrive si è occupato spesso di analisi dei siti che perpetrano attività di *phishing*. Nella fattispecie ho analizzato sia la classica tipologia di phishing realizzata con reindirizzamento verso un sito fantoccio che quella basata su email in codice HTML aventi già al loro interno tutto quanto necessario per permettere il furto delle credenziali all'utente.

Questi siti, come nel caso di una login ad un falso sito delle Poste Italiane, vengono diffusi attraverso link spediti via mail, che informano l'ignaro, e spesso incosciente, utente dell'imminente scadenza delle sue credenziali o che per causa di un guasto tecnico è richiesta cortesemente la reimmissione della propria username e della password, in modo da normalizzare la situazione.

Queste mail presentano sempre al loro interno una form HTML o al massimo un link ad un'altra pagina, predisposta opportunamente per sembrare a tutti gli effetti la pagina ufficiale delle Poste Italiane, nella quale l'utente dovrà procedere al login. Sovente per comprendere quali sono le alterazioni operate dai malfattori alla pagina ufficiale dell'ente è opportuno scaricare il codice HMTL della vera pagina comparando meticolosamente i link, i file e gli oggetti tra i due codici.

Per automatizzare il più possibile la comparazione si può usare il comando diff sebbene sia necessario considerare che anche la sola presenza di uno spazio in più nella formattazione di una linea rispetto alla sua omologa porta in output la segnalazione di linee differenti. L'opzione `-b` è utile in questo senso in quanto permette di ignorare le differenze prodotte dai soli spazi tra caratteri.

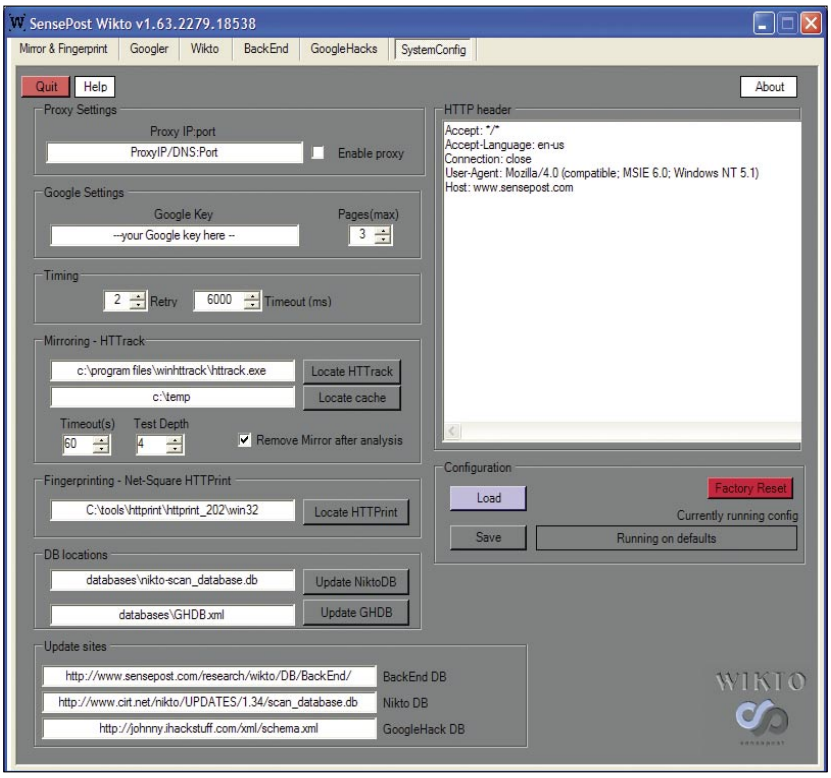

Fig. 11: *Wikto e sua integrazione con programmi esterni*

Non ci dedichiamo in questa sede a valutare gli aspetti di ingegneria sociale dietro ad una mail come questa, ci basti segnalare l'evidente tentativo di mettere a proprio agio l'utente con il comunicato indicando nella stringa dell'indirizzo: *VERIFICATO MEDIANTE TELEGRAMMA*. Ovviamente all'ignaro utente non sarà arrivato nessun telegramma dall'ente Poste, ma il solo fatto che questa verifica sia stata fatta, o meglio sia in corso d'opera, dovrebbe, nelle intenzioni del malfattore, rendere più fiducioso l'utente sulla bontà del comunicato e sull'attendibilità della fonte.

Nell'operazione di comparazione solitamente salta sempre fuori una stringa che riassume la sostanza del phishing, un qualcosa di questo genere (*L'esempio è una Proof of Concept* estratta da un caso reale, i riferimenti logici sono stati alterati per evitare la diffusione di dati sensibili.) Vedi Listato 4:

```
# diff -b fake.html real.html
```

Il risultato di questo controllo è un testo in cui ogni differenza è segnalata da un codice numerico xcy che segnala le coordinate all'interno del codice nelle quali si sono individuate le differenze, quest'ultime sono poi definite con il carattere (*c*) quando la stringa individua un cambiamento e (*d*) una cancellazione. Il carattere `< precede il testo del primo file, il carattere >` il testo del secondo file.

Di tutti i risultati che il comando *diff* riporta dall'analisi comparata tra sito falso e originale fa emergere una serie di dati che commentiamo di seguito:

```
8c6
< <link rel="stylesheet" href="http:
//www.***.it/online/
loginhome.fcc?[...]"
type="text/css" media="all" />
---
> <link rel="stylesheet" href="/lin/
css/lin2obj.css"
type="text/css" media="all"/>
```

Questa porzione di codice definisce lo stylesheet usato nella pagina. Per evidenti necessità di conformità all'aspetto del sito originale, il sito falso indirizza lo stylesheet con un link assoluto ai parametri definiti nel codice del sito originale. Nella successiva porzione di codice evidenziata troviamo:

```
31c26
< <a href="login.htm?requester=signon"
class="obibtn">Sign On</a>
---
> <a href=https://bancopostaonline.post
e.it/bpol/[...]/Controller?requester=s
ignon
class="obibtn">Sign On</a>
```

Questo passaggio mostra come nel sito originale ci sia un collegamento alla pagina di *Sign On* ufficiale che nel caso del sito falso è reindirizzata ad un *login.htm* certamente sospetto. Questa pagina contiene form HTML che richiedono le credenziali utente. Se approfondissimo l'analisi delle successive pagine post-autenticazione si noterebbero in maniera ancora più macroscopica le differenze tra un sito e l'altro.

Nell'ultimo troncone, nascosto agli occhi dell'utente, c'è un link ad un'immagine nascosta e grande 1 pixel x 1 pixel, la quale riveste la funzionalità di contatore dei contatti.

```
65c59
< <!-- COMMENT--></object>
[...]
< <IMG SRC="http://www.myspace.com/
[...]/pers?s=1001068&t=1101101461"
ALT=1 WIDTH=1 HEIGHT=1>
```

In effetti ogni utente che visualizzerà la mail di phishing con il proprio browser o con il proprio client di posta via web cercherà di scaricarsi in locale l'immagine contattando il link *http://geo.yahoo.com/serv?s=1001068&t=1101101461* dove risiede l'immagine. Questo permetterà al malfattore di identificare una sessione di consultazione del messaggio e di poter tracciare l'IP da cui proviene la lettura del messaggio.

Tool di Copia di un sito Web

L'uso del browser per visionare e valutare le singole pagine web è comodo, ma quando è necessario lavorare sull'intero sito, soprattutto in situazioni di incidente, è preferibile ricorrere ai tool che permettono il dowload in locale dell'intero ambiente. Questo è possibile attraverso software tra cui cito Wget e *HTTrack*.

Wget

Wget è un tool via shell di Unix che permette il download delle pagine web costituenti un sito. Lascio alle pagine del manuale il compito di illustrarne le funzionalità. È disponibile anche una versione per Microsoft Windows. Il tool in questione è scaricabile alla pagina *http://www.gnu.org/software/Wget*.

HTTrack

HTTrack è un'utilità disponibile in licenza GPL per Windows 9x/NT/2000/XP Linux e Unix/BSD ed è liberamente scaricabile dal sito: *http://www.httrack.com*.

Questo software permette il download di interi siti in cartelle locali riuscendo anche a ricostruire la struttura stessa dell'alberatura originale e salvando il codice HTML, le immagini e tutti i file inerenti il sito. *HTTrack* rimappa inoltre la struttura dei link relativi del sito originale permettendo di navigare tra le pagine come se l'accesso fosse via web e garantendo l'aggiornamento dei siti di cui si è già svolto un mirror e che hanno visto mutare la loro struttura rispetto all'ultimo accesso di *HTTrack*.

Il lavoro di rimappatura dell'albero di directory e collegamenti del sito può essere a volte scomodo per le operazioni di confronto tra versioni diverse, inoltre questa operazione, alterando il codice del sito inquina le prove e può rendere inammissibile per la legge il codice raccolto, ma è possibile aggirare il problema facendo analizzare il sito scaricato ad uno script Perl che metta in evidenza le alterazioni introdotte da *HTTrack* (predicibili e riproducibili). Inoltre dobbiamo considerare che la copia in locale occorre anzitutto per approfondire le indagini e poter svolgere maggiori inferenze sul sito e sul suo contenuto, inferenze che a volte risulta impossibile svolgere online per questioni di banda, di tempo a disposizione o anche di scarsa flessibilità da parte di alcuni tool di analisi del contenuto.

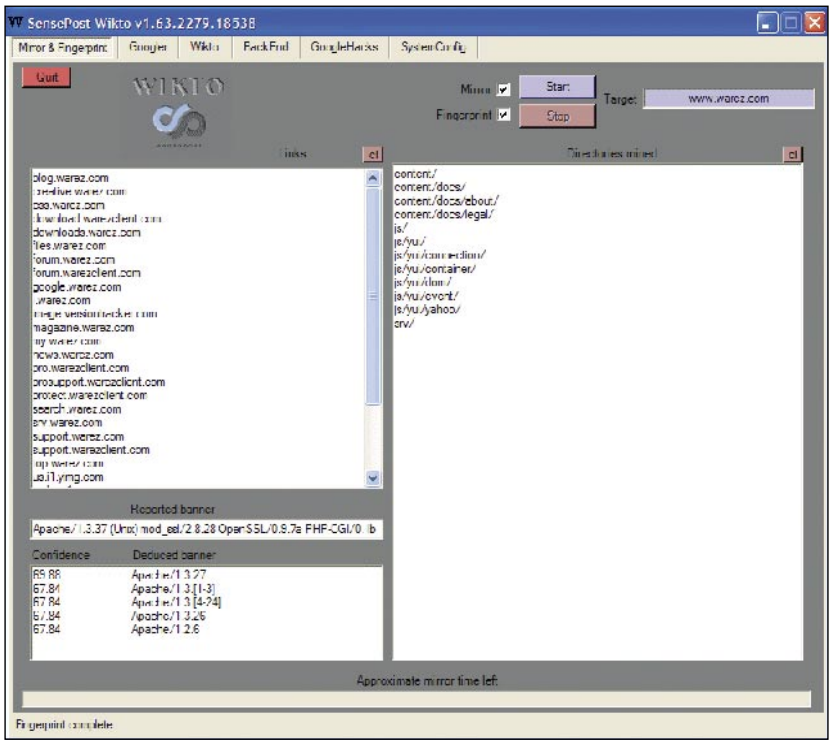

Fig. 12: *Wikto – Directory Listings e Fingerprint*

Copiare localmente l'intero Sito Web

Molti siti di cracker, virii e *phishing*/*pharming* hanno una vita breve nell'Internet, a volte poche ore, il tempo che il *Service Provider* o l'azienda che ospita il sito si renda conto che esso è fonte di attività illecite e che quindi lo spenga. Questo a volte danneggia l'attività di indagine. È perciò essenziale agire rapidamente in presenza di una segnalazione e in questi casi è fondamentale il supporto che ci offrono i tool di copia.

HTTrack o Wget sono strumenti ideali per la funzione di copia in locale di questi siti.

Chi scrive raccomanda però di fare attenzione alla quantità di spazio messo a disposizione per la copia (alcuni casi di siti cinesi hanno raggiunto svariate centinaia di Mb di spazio e in alcuni casi oltre 1 Gb) e alla banda richiesta per velocizzare l'operazione, qualche volta la copia è stata interrotta proprio a causa dello stop imposto dal provider al sito che ha impedito di terminare il lavoro. Con Wget basta eseguire il comando:

```
% Wget -m http://www.nomesito.com
```

Ricordiamo che di default Wget scaricherà le pagine presenti sul sito web e non seguirà i link agli altri siti, è possibile far eseguire la copia comprensiva dei vari rimandi e collegamenti attraverso l'opzione `-H`, ma è necessario tenere a mente le raccomandazioni precedenti.

È inoltre importante ricordare che, nel caso di Wget, tutti i link rivolti ad altre pagine impostati come link assoluti non funzioneranno se copiati nella maniera sopraesposta, per normalizzare questo aspetto e garantire la fruibilità di tutti i collegamenti è necessario impostare il seguente comando di copia:

```
# Wget -m -k http://www.nomesito.com
```

Lo scotto da pagare, in questo caso è dato dal fatto che la struttura HTML del sito copia non sarà comparabile, per questi aspetti, all'originale. Altro aspetto da ricordare è che per ogni pagina di codice HTML avente una directory collegata Wget salverà una variante del suo indice con il nome originale seguito da un'opzione, come di seguito:

```
index.html
index.html?D=A
index.html?D=D
index.html?M=A
index.html?M=D
index.html?N=A
index.html?N=D
index.html?S=A
index.html?S=D
```

Il motivo è dato dal fatto che Wget raccoglie e schematizza i dati in differenti modalità, ognuna indicata da un codice dopo il nome della pagina ovvero *index.html.?Y=A/D* dove *Y* è un valore semantico compreso tra D – M – N o S. Le differenti versioni sono organizzate per nome (*N*), data di ultima modifica (*D*), grandezza (*S*) e descrizione (*D*) in forma ascendente (*A*) o discendente (*D*). Tutte queste varianti non sono utili ai fini dell'analisi e ci possiamo concentrare sul file principale, nel nostro caso *index.html*.

Wget ha anche altre limitazioni nel suo funzionamento che lascio alla documentazione ad esso acclusa l'onere di evidenziare, ma ha anche alcuni vantaggi, come per esempio quello di scaricare anche le intestazioni Http del server Web su cui è pubblicato il Sito analizzato, un elemento che ci fa risparmiare tempo e su cui torneremo più avanti.

Con *HTTrack* occorre svolgere una serie di operazioni piuttosto immediate per avere in breve (dipende dalla banda e dalla grandezza del sito) una copia locale di un sito web.

Altre informazioni

Non tutto quello che è necessario per un'analisi forense di un sito web è fornito dall'analisi del codice HTML, come d'altra parte non tutto sulle meccaniche di un evento criminale può essere raccolto attraverso i dati raccolti dalla scena del crimine.

Molte altre informazioni essenziali possono essere raccolte, per esempio attraverso le intestazioni HTTP del server che ospita il sito analizzato, o ancora la data, gli orari, i log, i web cookies, tutto questo può essere raccolto solo attraverso un'operazione di analisi lato server.

Tra le varie operazioni di analisi che possono essere condotte lato Server una delle più utili è la cosiddetta *Directory Listings*, l'enumerazione e l'accesso alle cartelle del sito.

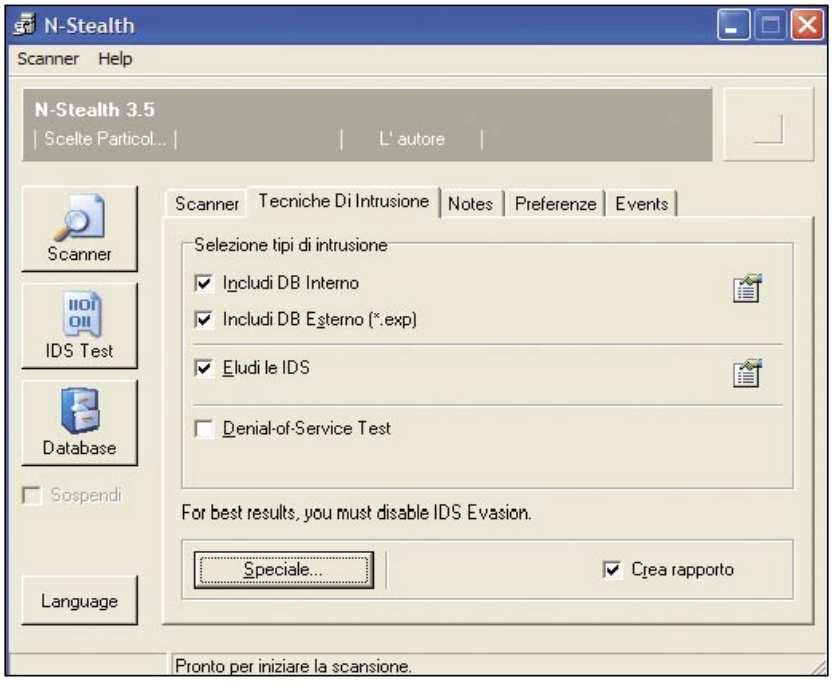

Fig. 13: *N-Stealth*

Directory Listings

Come tutti i servizi *Client-Server*, un sito web basato su HTTP lavora sul principio che se un client, sotto forma di browser, richiede un URL, il server stesso risponderà in uno dei seguenti modi:

- Restituisce la risorsa richiesta,
- Restituisce un errore,
- Restituisce una pagina html contenente l'elenco degli oggetti contenuti nella directory.

È proprio quest'ultima possibilità che andiamo ad approfondire.

Questa operazione, e cioè il chiedere ad un Server web una URL che produca come risposta la lista di una directory è chiamata *Directory Listings*.

La possibilità di lasciare all'utente la libertà di visualizzare il contenuto di directory e cartelle è limitabile o eliminabile attraverso policy e moduli configurando opportunamente il Server web su cui si installa il sito, in *Apache* per esempio viene gestita attraverso il modulo `mod_autoindex`. Tramite la direttiva *Options* è possibile abilitare e disabilitare la visualizzazione dei file contenuti in una directory, quando non esiste la *DirectoryIndex*, cioè un file predefinito come home page:

```
Options +Indexes
Options -Indexes
```

In *Internet Information Server* (IIS) si può abilitare/disabilitare attraverso un'opzione attraverso una classica spunta su un menù nel pannello di configurazione principale di IIS (vedi la Figura 10).

Per motivi di sicurezza e riservatezza è generalmente consigliabile disabilitare l'opzione di enumerazione delle cartelle, a meno che l'intento sia proprio quello di mostrare tutto il contenuto di una directory pubblicamente. Ma spesso, per pigrizia, incompetenza o scarso senso pratico, i vari passaggi necessari a garantire la protezione dalla *Directory Listings* non vengono effettuati dai *System Administrator* e questo comporta, per chi svolge analisi, un bel vantaggio in quanto si possono raccogliere informazioni e dati magari non pensati per essere pubblicati.

Questo è tanto più vero e possibile tanto più criminosi e truffaldini possono essere gli intenti del sito analizzato. Infatti normalmente un sito di phishing o anche di warez è realizzato in fretta e con l'obiettivo di rimanere attivo giusto il tempo necessario a rubare quante più credenziali possibili o a diffondere quel software o quello screener nei tempi più rapidi possibili e in questo senso non molte sono le attenzioni riposte sull'aspetto della sicurezza.

Questa possibilità è comunque molto utile per collezionare pagine grezze o non pubblicate o per avere un'ulteriore chiave di lettura per comprendere la consistenza, l'esperienza e la capacità delle persone dietro a questi siti.

Se poi siamo liberi di investigare più a fondo, possiamo arrivare ad indicizzare e consultare le *Directory Nascoste* (*Hidden Directories*).

Directory Nascoste

È molto semplice arrivare ad ottenere una vista delle cartelle pubbliche di un sito web, ma per converso ci sono cartelle, file e dati che possono non essere correlati attraverso un'analisi del codice HTML del sito. Queste cartelle, normalmente non visibili perché posizionate logicamente nell'area privata del Server, sono visionabili attraverso una serie di modalità tra le quali i tentativi diretti alle cartelle di default come per esempio */cgi*, */html*, */jsp* o, attraverso tools quali *Wikto*, *N-Stealth*, il vecchio *Whisker*, *Nessus*, ecc...

In UNIX, come è noto, le cartelle che iniziano con il punto, ad esempio: */home/userA/.mysecret/* non sono visibili se non utilizzando comandi quali ls `-a`. La stessa modalità è usata nei server Apache per nascondere queste cartelle ad occhi indiscreti, anche quando sia lasciata abilitata la funzionalità di *Directory Listings*. Questa caratteristica è stata spesso utilizzata da chi realizza siti di phishing/pharming per nascondere il contenuto dei propri file come ad esempio dei file legati all'intercettazione di account. È da evidenziare il fatto che il contenuto è protetto, solo i nomi delle cartelle e la loro visibilità è oscurata quindi, una volta individuate queste, è semplicissimo visionare i file in esse contenuti.

Per una più approfondita scansione di cartelle nascoste è essenziale utilizzare programmi adatti, tra questi consiglio:

- Nikto,
- Wikto,
- N-Stealth 3.5 (Free),
- Parosproxy.

Per altri tool si può consultare la pagina: *http://sectools.org/web-scanners.html*.

Come già indicato ho preso a riferimento solo i tool gratuiti. Per sintesi non dedico spazio a parlare in speci-

In Rete:

- Forensics:
  http://ftp.cerias.purdue.edu/pub/papers/steve-weeber/spaf-weeber-forensics.pdf,
  http://johnny.ihackstuff.com/,
  http://www.seclab.tuwien.ac.at/papers/pixy.pdf.
- Honeynet Project:
  http://www.honeynet.org/.
- Phishing:
  http://www.honeynet.org/papers/phishing/,
  http://www.antiphishing.org/crimeware.html,
  http://www.sicurezzainformatica.it/archives/phishing_e_truffe/,
  http://www.anti-phishing.it/news/articoli/news.03012007.php,
  http://www.netcraft.com.

fico di questi tools lasciando ai link in appendice e alle schermate successive il compito di descrivere brevemente questo aspetto (Per questioni di privacy ho alterato le indicazioni degli indirizzi IP dei siti analizzati).

Nikto

Per descrivere questo tool userò le stesse parole dei suoi creatori: Nikto è un web scanner Open Source (GPL) che permette di svolgere tests verso web servers per varie tipologie di elementi inclusi più di 3200 files/CGI a rischio. [...] gli elementi di scansione e i plugins sono frequentemente aggiornati/aggiornabili, anche automatizzandone le procedure.

Questo software non è particolarmente silenzioso nel suo test ed è molto facile vederne l'attività nei log di un server analizzato o ancora più evidente risulta in presenza di un IDS.

Nikto svolge il suo lavoro di test con l'obiettivo di segnalare vulnerabilità applicative e del Server. Non usa queste vulnerabilità per rendere possibile la presa di possesso di shell nella macchina, ma solo con l'obiettivo di informare il tester dello stato di protezione del Sistema. È però chiaro che ad un'analisi svolta con *Nikto* può seguire un'attività di penetrazione o di raccolta di dati attraverso l'uso delle informazioni che il software ha collezionato. È da rammentare che *Nikto* è piuttosto rumoroso e lascia tracce evidenti nei log del server analizzato, a meno che non si usino plug-in esterni.

Wikto

Uno dei software più recenti ed efficaci nell'analisi di un Web server. Si basa, come il nome fa pensare, su Nikto (e sul suo vasto database) ampliandone però la configurazione e lo scenario di utilizzo in quanto si interfaccia anche con altri programmi tra cui *HTTrack* e *HTTprint*. Altra funzionalità molto interessante è il mining delle cartelle e dei file attraverso *Google*. Si può scaricare da: *www.sensepost.com/research/wikto/*.

Vale anche in questo caso la segnalazione della *rumorosità* di questo software nella scansione.

N-Stalker Free Edition

Un tool gratuito che in realtà è una versione ridotta di *N-Stalker Infrastructure Edition* un software commerciale. Nonostante la versione *Free* sia ridotta nel database di signature con cui svolgere i test, contiene comunque circa 18,000 signatures e controlli di Sicurezza per un sito Web.

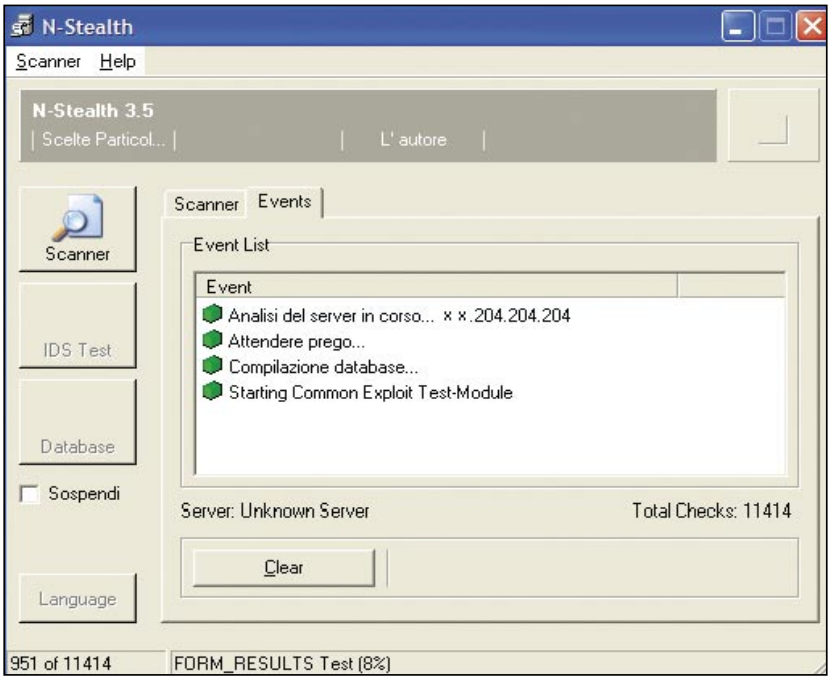

Fig. 14: *Analisi Sito web via N-Stealth*

È ancora tra gli strumenti più validi di controllo per Microsoft Windows. Si può scaricare da: *http://www.nstalker.com/products/free/*.

Parosproxy

Scritto (in Java) da alcuni professionisti della Sicurezza e disponibile gratuitamente, questo tool di analisi e di test è veramente molto potente e ben organizzato.

Permette molte attività di analisi per sessioni HTTP/HTTPS.

Funziona sulla base del principio del proxy applicativo (su porta 8080) per le sessioni web, tra un server e il proprio browser Internet opportunamente configurato (proxy su 127.0.0.1:8080) e così organizzato permette di intercettare e alterare le sessioni compresi i cookies e le form.

È un tool che merita un approfondimento specifico in quanto ha molte funzionalità interessanti tra le quali quella degli *Spider*: *http://www.parosproxy.org*.

I tools indicati sono utili per molte attività non solo di *Cyberforensics*, ma ad esempio di *Pen Testing*. Cosa cerchiamo attraverso questi tools, anzitutto ulteriori caratteristiche del Sito Web analizzato, quali vulnerabilità e cartelle nascoste.

Le vulnerabilità ci possono dare indicazione della qualità del lavoro di protezione svolto sul sistema e quindi dell'attenzione riposta nel mettere in linea il sito in questione. Le cartelle nascoste sono invece l'obiettivo principale di questa fase dell'indagine, come vedremo più avanti in queste cartelle si possono nascondere informazioni molto utili all'attività di indagine, come per esempio account utente, log o altri dati altrettanto preziosi.

Un altro tool altrettanto importante di quelli sopra menzionati è il comune *Motore di Ricerca*.

Cercare le cartelle nascoste attraverso i motori di ricerca

Accennavo nell'introduzione dell'articolo al fatto che fino a qualche tempo fa si poteva svolgere un mass defacement attraverso l'uso dei motori di ricerca. Questa pratica nasce

dal fatto che gli spiders di un motore di ricerca, nella loro incessante attività di scoperta e indicizzazione riportano al *Database Generale* anche directory e file nascosti nelle pieghe dei siti web, o semplicemente permettono lo scavenging delle cartelle e dei file arrivando spesso a fornire il path assoluto di pagine intranet o peggio delle aree di upload utilizzate per aggiornare e mantenere un sito.

Questo porta in breve a poter sostituire senza troppe difficoltà la pagina originale dell'index o di altre porzioni di un sito, con altre pagine e altri contenuti (il defacement, appunto).

Per individuare file all'interno di un sito web è sufficiente, utilizzando *Google* per esempio, svolgere una ricerca in questo modo:

```
keyword site: www.nomesito.com -
filetype: asp
o anche
welcome to intranet
```

Indovinare i nomi più comuni per queste ricerche porta sempre a risultati molto interessanti, rimane la raccomandazione etica che deve fermarci laddove stiamo violando la privacy e la legge.

Questo processo è una sorta di tentativo alla cieca, ma dobbiamo anche ricordare che alcuni software utilizzati per creare siti web creano cartelle di default e file con nomi ed estensioni note permettendo ricerche anche piuttosto ben mirate. Esempio che possiamo riportare sono i seguenti:

```
css - una posizione standard per gli
stylesheet files
javascript, js - la posizione standard
per salvare i file JavaScript
_vti_cnf, _vti_bin, _vti_log, ecc. -
Cartelle create da Microsoft FrontPage
_notes - File XML creati da Macromedia
Dreamweaver
```

Queste cartelle, cercate attraverso il modo sopra indicato, possono portare decine di migliaia di risultati con *Google*, *Altavista*, *Yahoo* e gli altri portali. C'è poi da considerare che la presenza di un determinato tipo di file può indicare una vulnerabilità del sito o la possibilità di svolgere su di esso determinati e specifici attacchi.

Un modo per assicurare che gli spiders o i web crawler non indicizzino queste cartelle è legato all'uso del file *robots.txt* che, laddove si inseriscano i nomi e i percorsi delle cartelle da nascondere, permetterà di far sfuggire tali cartelle all'indicizzazione. In realtà però questo comporta il rischio di esporre, attraverso il file, proprio i nomi e i percorsi delle cartelle che non si vuole siano pubblicamente accessibili portando al risultato esattamente contrario. Per capire questo basta provare a svolgere una ricerca su *Google* in questo modo:

```
inurl:robots.txt filetype:txt
```

Conclusioni

Questo breve excursus nella *Cyberforensics* mi auguro che possa stimolare ulteriori approfondimenti che per ovvie ragioni di spazio non sono stati possibili qui.

Attualmente la tematica è molto calda, sia per il suo legame con la lotta al *Phishing*/*Pharming*, sia per i suoi connotati legali e per i risvolti legati alla privacy e alla protezione dei dati e dell'accesso alle informazioni aziendali.

Proprio in questo periodo si assiste al mutamento delle modalità di distribuzione dei messaggi di Phishing. Questo è il segnale più evidente del fatto che, proprio grazie alle investigazioni finora condotte e ai risultati raggiunti questa piaga era stata limitata, nel suo propagarsi. Ormai molti dei principali sistemi Antispam e antivirus a disposizione degli utenti e dei *Service Provider* possono individuare, analizzando il testo del messaggio HTML, un tentativo di *phishing* via email (principale vettore di queste truffe).

Ma la lotta continua... le nuove modalità di *Phishing* contemplano la distribuzione di messaggi in *Adobe*/*Macromedia Flash*, un formato multimediale che è ormai uno standard di fatto supportato da tutti i browser, che per sua costituzione non permette ai filtri integrati nei browser, o quelli esterni come gli antispam o gli antivirus, l'analisi del contenuto del messaggio.

Né consegue che questo fenomeno, come pure il fenomeno dei *Defacement* o della distribuzione illegale di contenuti multimediali, o peggio quello riprovevole della pedopornografia, necessitano di sempre nuovi e motivati professionisti sia sul campo che nella ricerca. E ricordate, come sostiene Gil Grissom: *Nessun ritrovato tecnologico potrà mai superare il caro vecchio cervello umano.* ●

Fonti Documentali:

- *http://www.openskills.info*,
- *http://www.risorse.net*,
- *http://www.securityfocus.com*,
- *http://it.wikipedia.org/wiki/Pagina_principale*,
- *http://www.owasp.org/index.php/Main_Page*,
- Apache Security di Ivan Ristic, Oreilly 2005,
- Internet Forensics di Robert Jones, Oreilly 2005.

Cenni sull'autore

Stefano lavora nell'IT da dieci anni ed ha raccolto in questo tempo una vasta serie di esperienze in progetti internazionali in ambito di Storage, Networking, Wireless e Sicurezza. In Italia ha collaborato con molti *Service Provider* e con alcune tra le maggiori aziende nazionali negli ambiti del Networking e della Sicurezza.