Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Informatica

SISTEMA SERVER-SIDE DI ACQUISIZIONE FORENSE DI CONTENUTI WEB ASINCRONI

Relatore:
Chiar.mo Prof. Fabio Vitali

Presentata da:
Massimiliano Dal Cero

Correlatore:
Dott. Stefano Fratepietro

II sessione
(20 dicembre 2017)
Anno Accademico 2016/2017

Dance like no one's watching.
Decrypt like everyone is.

Indice generale

Introduzione

Il Web è oggigiorno uno "*spazio digitale*" molto complesso, in grado di creare un panorama in continua evoluzione, sia per le tecnologie in esso presenti, sia per le dinamiche delle relazioni umane (legislative e comportamentali).

Il panorama delle tecnologie e dei modelli architetturali delle applicazioni Web è cambiato molto negli ultimi anni. Le nuove tecnologie e i vari framework permettono di realizzare applicazioni ad alto valore aggiunto, nelle quali il livello di interazione con l'utente è notevolmente aumentato rispetto a una classica applicazione Web della generazione precedente. Dal giorno della sua prima definizione (ad opera di Macromedia nel 2002) il termine di "*Rich Internet Application*" (RIA) è divenuto uno dei più usati e abusati sul Web; da allora nuove tecnologie e nuovi paradigmi si sono avvicendati sullo scenario con l'obiettivo di aumentare la potenza e l'interattività delle GUI (*Graphic User Interface*) delle applicazioni al fine di rinnovare e migliorare la cosiddetta "*user experience*". Nello scenario appena introdotto il **paradigma asincrono** risulta essere il massimo esponente di questa evoluzione, che è stato in grado di eliminare la natura primitiva del Web contraddistinta da momenti alternati di "*Request-Response*", introducendo un intermediario (in genere un motore javascript) tra l'utente e il server in grado di creare un'interazione continuativa.

Partendo dall'idea originaria di Web proposta da Tim Berners Lee, la maggior parte del lavoro fatto nell'area delle "*Web application*" ha avuto come obiettivo quello di far

evolvere il modello **statico** del WWW della prima era verso un meccanismo **dinamico** e "*application based*" che, con l'introduzione di *AJAX* ("*Asynchronous JavaScript and XML*"), ha permesso la creazione di applicazioni realmente asincrone (non solo per certe operazioni di refresh grafico ma anche per le invocazioni di operazioni di business).

Il presente elaborato nasce pertanto dalla sentita necessità, proveniente dal mondo dell'*informatica forense* (anche detta "*digital forensics*"), di riuscire a catturare in modo completo, attendibile e affidabile tutte quelle prove digitali (*e-evidence*) veicolate a mezzo del Web moderno e della sua relativa evoluzione asincrona.

Scopo del progetto di tesi, descritto nelle pagine seguenti, sarà quindi quello di creare uno strumento forense di acquisizione di contenuti Web asincroni e/o comunque di articolata fruizione, in grado di aggiungersi al panorama presente (sia forense che tecnologico) potendo offrire caratteristiche ad oggi ancora inespresse da altri prodotti con intenti simili.

Quando si parla di "contesto forense" in informatica, si introducono una serie di aspetti che influiscono pesantemente sia sulle procedure che sui software utili a far emergere le evidenze digitali. Come appena espresso, vi sono 3 aspetti che un software forense deve soddisfare: esso deve essere **accurato**, **attendibile** e **completo**. [FG2013]

- **Accurato:** dovrà seguire le più attuali procedure riconosciute e adottate dalla disciplina dell'informatica forense, in grado di fornire processi chiaramente definiti, verificabili e documentati.

  Dovrà quindi essere possibile fornire l'informazione minima necessaria tale da rendere esplicitamente chiare tutte le procedure svolte dal software utilizzato, permettendo in tal modo al consulente tecnico di documentare minuziosamente sia l'attività svolta che il software adottato, al fine di fornire il necessario ed adeguato rigore scientifico da affiancare al relativo procedimento giuridico.

- **Attendibile:** mediante appositi strumenti hardware o software si dovrà garantire l'esatta corrispondenza con il contenuto originale, potendone assicurare la sua integrità, sia nel tempo (memorizzazione) che nello spazio (trasmissione).

  Il software utilizzato in un contesto forense dovrà avere caratteristiche aggiuntive (spesso confuse con eccesso di zelo) in grado di dare le dovute garanzie sul contenuto acquisito e relativa integrità e genuinità.

  Oltre alla firma hash del singolo dato che ne garantisca la sua integrità, risulterà ulteriormente necessario tenere traccia di tutte quelle informazioni "laterali" che hanno accompagnato lo svolgimento dell'operato, come per esempio il log dettagliato delle attività svolte e i relativi metadati temporali. Nello specifico, calandosi nel contesto del presente progetto, risulterà inoltre necessario fornire il tracciamento di tutti gli header HTTP, il traffico di rete generato con successiva possibilità di interpretare anche tutto il traffico veicolato su protocollo HTTP**S**, la registrazione della sessione video, etc.

- **Completo:** è quantomai importante non perdere prove digitali di interesse durante le attività di acquisizione. Risulta altresì fondamentale non confondere il concetto di **completezza** con quello di **totalità** (non necessaria né richiesta) in qualunque forma essa si possa presentare nel contesto applicativo di interesse. Anzi, ottimale sarebbe evitare le cose superflue, acquisendo solo quanto realmente pertinente al quesito in esame. Ovviamente questo non è sempre possibile, si pensi per esempio all'acquisizione fisica di un drive (esterno, interno o virtuale), che per sua specifica natura necessiterà del bitstream completo dal primo all'ultimo bit. Seppure in una specifica attività potremmo non avere la possibilità di scindere i contenuti di interesse da quelli che invece non lo sono, sarà compito del consulente tecnico valutare puntualmente gli elementi da mettere sotto acquisizione.

Nello specifico caso della presente tesi, sarà quindi necessario avere uno strumento software in grado di ottenere l'**informazione (completa) essenziale allo scopo**. Non sarà quindi necessario svolgere un'acquisizione completa di un "sito", di un "profilo" social o di ogni specifica componente presente in una data pagina, come per esempio video pubblicitari o quei contenuti che si caricano progressivamente allo scorrimento.

Risulterà decisamente più importante la completezza piuttosto che la totalità, con ben chiaro il primario obiettivo di giungere **puntualmente** al contenuto di interesse senza tralasciare informazioni ed altri contenuti utili al caso.

Oltre alla parte tecnica esiste inoltre la componente antropologica, che si manifesta sia nel ruolo di utente fruitore-creatore di contenuti sia in quello di legislatore.

Il ruolo umano contribuisce a rendere il Web un luogo di contenuti che potranno essere più o meno leciti in base alle singole normative vigenti. Per comprendere al meglio il contesto giuridico risulterà quindi importante comprendere anche tutti quegli aspetti caratterizzanti i comportamenti di interesse che ciascun individuo pone nei confronti del Web. È frequente infatti che i comportamenti tenuti per mezzo indiretto (quindi con la quasi totalità degli strumenti digitali) siano caratterizzati da atteggiamenti inconsapevolmente "irrequieti" in grado di esporre ciascuna persona a conseguenze gravi, (non solo legali), sia come vittima che come attaccante. Non di rado, infatti, certi atteggiamenti possono porre il singolo utente in situazioni tali da renderlo vittima di azioni illecite, indotte da comportamenti di "estrema fiducia" o di "estrema superficialità" che difficilmente terrebbe nel mondo fisico. Ci sono poi, dalla sponda opposta, quei comportamenti **illeciti** che il singolo soggetto, sia per buon senso che per propria educazione, non metterebbe mai in pratica nella vita sociale di tutti i giorni.

Tutti questi aspetti, congiuntamente al preesistente ordinamento giuridico italiano, hanno contribuito a far evolvere il contesto normativo parallelamente all'evoluzione tecnologica. Ma anche a fronte di una naturale, quanto inevitabile, evoluzione del

diritto, non aiuta il fatto che l'informatica stessa abbia percorso (e continua a percorrere) passi da gigante, trasformandosi in una sorta di bersaglio mobile impossibile da ingabbiare in regole e procedure fisse. In queste condizioni è difficile riuscire a mantenere il diritto e la prassi allineati all'evoluzione della tecnica, rendendo l'informatica forense (pure con la presenza di importanti trattati internazionali) tutt'ora priva di un rigoroso inquadramento tecnico-giuridico e sofferente della mancanza di protocolli e linee guida unanimemente condivise dalla comunità internazionale di esperti.

Nelle fasi di analisi, studio e nella successiva realizzazione del progetto di tesi, si tenterà di dare risposta a tutte quelle necessità appena esposte, con il parallelo tentativo, non meno importante, di creare uno strumento che risulti di facile utilizzo per l'utente/operatore. Per giungere a tale risultato si cercherà di ricreare un'ambiente già ben noto all'utente, così da abbassare al massimo la ripidità della curva di apprendimento, potendo lo stesso sfruttare conoscenze pregresse. Nello specifico, si valuteranno e studieranno tutte quelle soluzioni che permettano o agevolino la realizzazione di un prodotto in grado di **replicare l'utilizzo di un "normale" browser Web**, con la particolarità di essere annidato all'interno di un classico software per la navigazione Internet utilizzato dagli utenti. Si creerà quindi quella **metafora qui denominata "meta-browser annidato"** sulla quale si baserà il "***Forensics Browser***", goal finale del presente progetto di tesi.

Nel seguito del documento saranno trattati, prima di tutto, temi in grado di dare ampio discernimento sul tema "*acquisizione **forense** di un contenuto We*b". Si partirà quindi da una **prima fase di studio e di ricerca** che porrà l'attenzione su come l'evoluzione tecnologica abbia influenzato le dinamiche della società moderna, e come questo abbia inevitabilmente influenzato il panorama giuridico che per sua natura ha lo scopo di regolamentare la società.

Dopo un primo studio sullo stato dell'arte del panorama sociale e giuridico, **si proseguirà quindi con la vera e propria fase sperimentale**. Questa seconda parte avrà

come scopo la **realizzazione** di uno **strumento di acquisizione forense di contenuti Web**, che vedrà anzitutto lo **studio** e le **valutazione** delle attuali tecnologie con finalità simili, con l'obiettivo di individuarne mancanze, criticità ed eventuali migliorie (o funzionalità) ad oggi ancora inespresse. Successivamente alla predetta fase vi sarà la **realizzazione concreta** degli obbiettivi prefissati e verranno illustrate le fasi di sviluppo, cogliendo l'occasione per esporre le tecnologie utilizzate, snippet di codice rilevanti, diagrammi, screenshot, etc.

1 Mediasfera ed Evoluzione tecnologica: influenza sulle dinamiche della società e conseguenze giurisprudenziali

1.1 Società e mediasfera

Ormai il quotidiano vivere della nostra società porta costantemente ogni cittadino a fronteggiare contenuti digitali senza saperne valutare l'impatto che questi hanno nel mondo fisico e spesso si diventa vittima di essi, sia in maniera diretta che indiretta [CCCD2016].

Da un lato ci sono quelle persone che ancora non hanno capito che le azioni compiute a mezzo informatico hanno una reale ricaduta a livello giuridico, sia civile che penale, dall'altro ci sono i soggetti che subiscono questi atteggiamenti; si pensi banalmente, per esempio, alla ormai triste e diffusa abitudine di commentare sui social network riversando rabbia, calunnie e minacce, ovvero atteggiamenti che difficilmente una persona riporterebbe nel mondo fisico, ben cosciente delle ricadute che questi comportamenti avrebbero sia a livello legale che personale.

Il tutto viene oggi ancora più aggravato dalla pervasività dei contenuti digitali che sono arrivati a creare una vera e propria "*Mediasfera*" [RS2012] in grado di avvolgere le persone durante il proprio vivere quotidiano.

La fase storica attuale è caratterizzata da un'ubiquità dei *media* che non ha precedenti nella storia. Per riprendere il termine che si è appena introdotto, siamo immersi in permanenza nella *mediasfera*. I *nuovi media*[1] di varia natura, personali e no, sono infatti ovunque: addosso alle persone, per le strade, nei posti di lavoro, sui mezzi di trasporto, negli spazi pubblici e privati, nei negozi, nelle stazioni, negli ospedali, nelle banche... Ciascuno di noi ne ha addosso uno o più, siano essi materiali (hardware) o immateriali (software): da una parte telefonini, computer connessi in rete, tablet, fotocamere, webcam; dall'altra le applicazioni che possono girarci sopra, dai social network ai diversi programmi per i più svariati scopi. Questa rete ricopre l'intero pianeta: non c'è praticamente alcun punto geografico che non possa essere raggiunto dalla rete.

Ciò significa anche che ogni aspetto della nostra vita è ormai completamente pervaso da questa *mediasfera* (o mondo digitale) e di conseguenza di dispositivi di ogni genere che ci permettono di ricevere ogni tipo di contenuto come anche di produrlo e renderlo fruibile a livello globale. La convergenza di più *media* nello stesso supporto ha accentuato il fenomeno in misura drammatica. [RS2012].

Queste trasformazioni sono arrivate a urtare pesantemente anche quella che, col termine coniato da Pierre Teilhard de Chardin, possiamo chiamare la *noosfera* [PT1995] , cioè l'insieme dei pensieri, valutazioni, opinioni, concezioni sui temi più diversi, che risiedono nella testa dell'essere umano. Sono numerosi i fattori che modificano la noosfera, alcuni dei quali sono oggetti fisici e tecnologie: basta pensare a come il frigorifero, l'automobile e l'aeroplano hanno trasformato la nostra vita e la nostra visione delle cose. Ora intervengono in modo prepotente i *media* elettronici e telematici: la mediasfera.[RS2012]

D'altro canto, alcune categorie cruciali dell'esperienza interiore sono intaccate senza rimedio, per esempio la ricerca scientifica, che comincia a occuparsi di alcune dimensioni della mediasfera, fornisce informazioni più dettagliate e perturbanti: per

1 *Espressione entrata a far parte del lessico degli studi sulla comunicazione verso la fine del 20° sec., che indica i mezzi di comunicazione informatizzati [...] Sono considerati n. m. anche i videogames, oltre a tutto il complesso di applicazioni nate su Internet – dalla posta elettronica alle chat room, dal web ai forum, dai blog ai social network – e ai dispositivi utilizzati per accedere alla rete, quali i palmari, gli smartphone, i tablet.*

esempio, uno studio di psicologia sociale del 2011 [ZF2011] indica che usando l'email si tende a mentire molto di più rispetto a quando si comunica faccia a faccia. Questo dato di fatto collima con l'impressione che in generale, usando i media, ci si lasci andare a un linguaggio molto meno controllato e composto e si scelga facilmente un codice aggressivo e volgare. Può darsi quindi che internet possa influenzare negativamente la nostra personalità, comportando di conseguenza potenziali rischi "legali", i quali possono diventare ancora più seri se consideriamo che si fa un utilizzo sempre più "ossessivo-compulsivo" delle risorse digitali che gravitano nella mediasfera, che porta l'utilizzatore a comportamenti sempre più maniacali [RS2012].

Inoltre, il mondo digitale (che è bene non chiamare virtuale) e gli strumenti ad esso associati, non solo sono parte integrante della propria vita personale, ma toccano oramai anche settori professionali una volta ben lontani da essi [IQ2013]. Questa stretta vicinanza tra mondo digitale e praticamente qualsiasi aspetto delle attività del mondo fisico arriva ad influire non di poco anche nei procedimenti legali, infatti la presenza di un consulente tecnico d'ufficio (CTU) è ormai prassi nelle aule di tribunale per individuare le evidenze informatiche che saranno poi parte integrante del processo stesso, e che spesso risultano fondamentali alla risoluzione del caso [CCCD2016].

Volendo portare un esempio per meglio chiarire questo ultimo punto, un evento che dovesse riguardare un triste e rovinoso crollo di una struttura abitativa, richiederà non solo perizie sul campo da parte di ingegneri civili o architetti, ma anche la co-partecipazione di un "informatico" (termine purtroppo sempre abusato) in grado di estrarre evidenze dalle memorie digitali e di fornire ancora più chiarezza sulla questione, potendo analizzare per esempio l'evoluzione dei progetti (ormai tutti svolti con strumenti informatici), le comunicazioni (chat, email) tra le parti in causa, i contratti, immagini e video, etc. Di tutto il materiale acquisito e periziato è inoltre spesso necessario produrre un indicizzazione da mettere disposizione degli inquirenti così che possano successivamente cercare e trovare autonomamente (o quasi) le risultanze di interesse.

1.2 evoluzione tecnologica: Cloud e “Web 2.0”

“*L'evoluzione tecnologica continua a sfidare il processo*” - dice Gerardo Costabile Presidente di IISFA. Si vede quotidianamente come nella pratica giudiziaria la cultura verso la prova digitale e la doverosa adozione delle procedure necessarie a garantire la genuinità delle stesse sia ben lontana dall'essere assorbita, trovandoci costantemente in ritardo nei confronti delle innovazioni tecnologiche che continuamente irrompono nell'ordinamento figurando problematiche che saranno di difficile gestione da un punto di vista normativo e operativo, considerando anche la dimensione globalizzata di questi fenomeni che nel carattere nazionale dei sistemi processuali, specie penali, non trovano corrispondenza. [CCCD2016]

Ma se la pratica giudiziaria non tiene sempre il ritmo, così non è per il cyber-crime organizzato o per quello più "modesto" perpetrato da cittadini “normali”; entrambe le categorie hanno a “portata di cloud” un numero potenzialmente illimitato di informazioni e strumenti e possono quindi ricorrere al computer ed agli apparati di comunicazione digitale per commettere reati con grande facilità, potendo sperimentare modalità inedite di consumazione delle attività illecite, anche “tradizionali” (vedi ad esesmpio lo *stalking*), seppur lasciandone digitalmente traccia. E se risulta ormai imprescindibile, in qualsiasi tipologia di indagine, acquisire, analizzare e conservare tali tracce accedendo direttamente ai “dispositivi” utilizzati nella attuazione dell'illecito, quindici anni dopo la Convenzione di Budapest (i cui obiettivi sono ancora lungi dall’essere pienamente realizzati) nuove ed importanti questioni indotte dal "cloud" si profilano all’orizzonte.

Lo stesso canone fondamentale del processo accusatorio è messo in difficoltà dalla dimensione intangibile del cloud. Le prove acquisite in fase di indagine giungono al processo già cristallizzate e alla difesa, nell’impossibilità di concorrere alla loro formazione in contraddittorio, non resta che sostanziare il dibattimento alla stregua di un’analisi critica dell’attività inquirente. [CCCD2016]

Risulta quindi chiaro quanto sia importante fornire strumenti in grado di "congelare" le prove digitali, e se questo può risultare abbastanza "facile" per contenuti presenti su dispositivi fisici, dei quali si può effettuare un *immagine* con strumenti di acquisizione forense ormai consolidati, non è invece così ovvio (ne banale) quando il "congelamento" deve essere fatto su contenuti presenti sul *WWW;* Quest'ultimo scenario è ben noto per essere soggetto a continue quanto frequenti modifiche, spesso non sottoposte ad opportune pratiche di mantenimento dei log sui sistemi ospitanti, ubicati in Stati con regolamentazioni anche molto differenti da dove il caso dovrà essere sottoposto a giudizio. [CCCD2016]

Internet o anche detto "*spazio cibernetico*" ha quindi aperto certamente nuove strade per lo sviluppo economico, ma la digitalizzazione delle comunicazioni e lo scambio di informazioni porta con se il pericolo di furto, contraffazione e alterazione dei dati nello spazio cibernetico che può, al tempo stesso, essere altresì il luogo ideale per l'occultamento delle tracce e dei proventi dei reati. Alla luce della proliferazione della criminalità informatica e nel contesto di una continua evoluzione tecnologica a cui si accompagna una delocalizzazione dei dati e delle informazioni, appare indispensabile riflettere sui mezzi e gli strumenti che possano permettere all'autorità giudiziaria un efficace contrasto alla criminalità che si avvale dello strumento informatico e della "rete" in particolare; in questa ottica certamente assume un ruolo centrale la tematica dell'individuazione della prova elettronica (c.d. e-evidence) e dei mezzi di ricerca e acquisizione della stessa in un contesto, quello digitale, ontologicamente rarefatto a cui con difficoltà possono adeguarsi i tradizionali paradigmi dell'attività investigativa. Innanzi a questo scenario anche il Consiglio dell'Unione Europa ha recentemente messo al centro della propria agenda il problema della prova elettronica e della capacità di acquisizione della stessa.[CCCD2016]

1.3 Virtuale & reale

Si può notare come fino ad ora non si è mai chiamato "virtuale" il mondo che si trova "oltre allo schermo", etichettandolo invece con il termine "digitale" o "mediasfera",

così come non si è mai chiamato “reale” il mondo dalla parte che si trova a fronte dello schermo. Questa scelta nomenclativa non è di secondaria importanza [U1,IQ2013] e sarà quindi mantenuta in tutto il documento, così da poter definire in maniera chiara delle terminologie che non arrechino malsane interpretazioni dato che, contrapponendo tra loro il “virtuale” e il “reale”, si corre il rischio di rafforzare l'idea che tutto quello che avviene per mezzo di dispositivi digitali (siano essi computer, smartphone o qualsiasi altra tecnologia) non abbia una connotazione reale, degli effetti e delle vere e proprie responsabilità giuridiche in calce a chi compie una determinata azione. “Entrambe le realtà sono reali”, ed in entrambe le realtà esistono responsabilità dirette e personali, che a fronte di azioni più o meno gravi possono portare l’individuo a dover confrontarsi con la legge dello Stato. [U1,IQ2013]

1.4 Conclusioni

A fronte del panorama appena esposto, lo scopo del presente progetto di tesi sarà realizzare un sistema di acquisizione forense in grado di adeguarsi agilmente allo scenario digitale, che possa quindi rendersi utile per far fronte al più ampio numero di casi possibile e oltrepassare le barriere (tecniche) che il web spesso pone nell’accesso al dato.

Nei prossimi due capitoli si tratteranno casi legali e questioni tecniche, così da aiutare il lettore a meglio comprendere il variegato, quanto intricato, panorama nel quale ci dovremo introdurre.

Conscio che il presente documento non è il luogo più idoneo per poter fornire un quadro esaustivo, nel capitolo 2 si tenterà comunque di mostrare una panoramica degli aspetti legali, raggruppandoli per casi di interesse, e nel capitolo 3 si fornirà invece un quadro del panorama tecnico dove si coglierà l'occasione per trattare le particolari tipologie di contenuto e i relativi ostacoli di accesso alle informazioni.

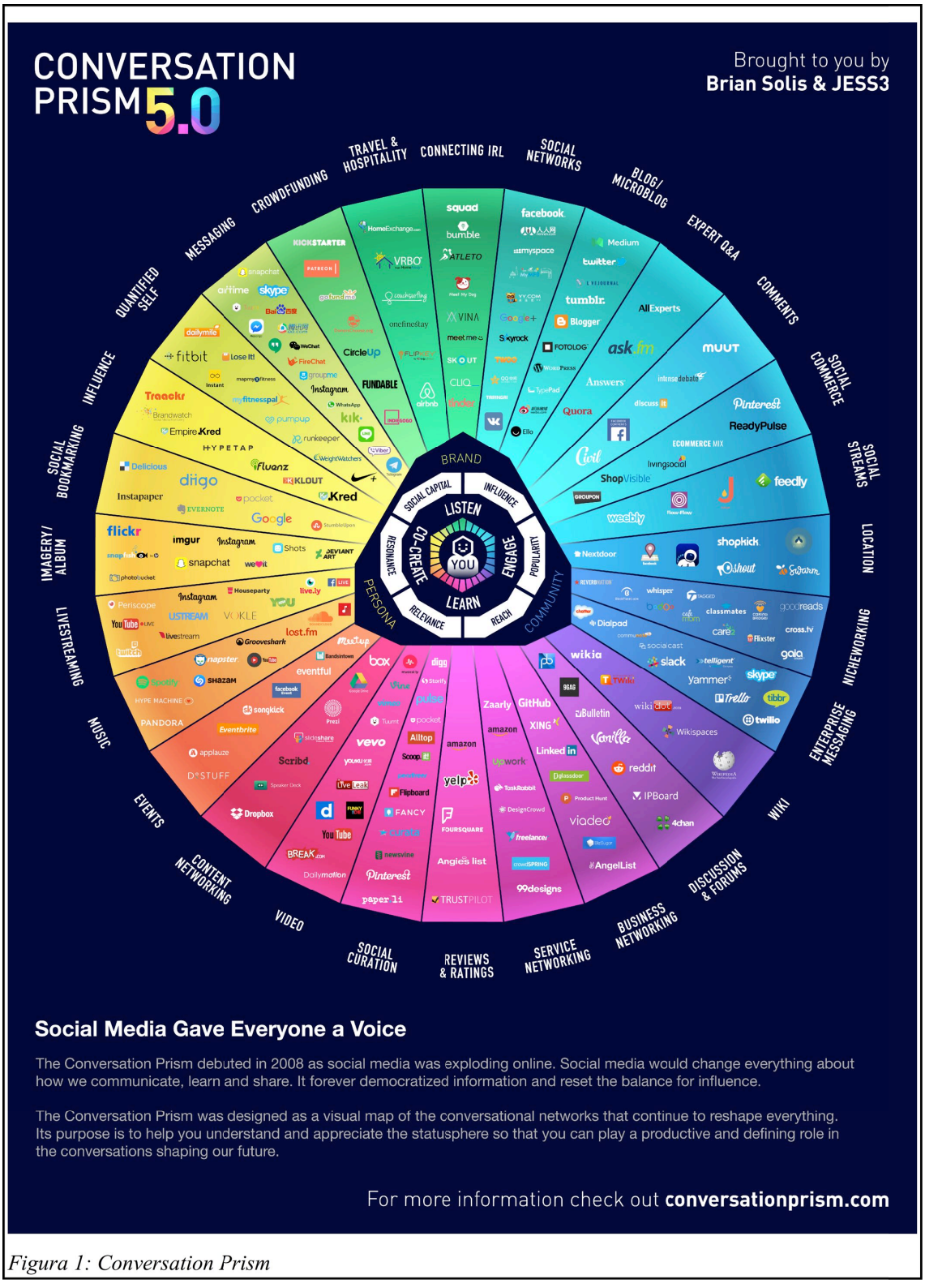


Figura 1: Conversation Prism

Lanciato nel 2008 da Brian Solis e JESS3 (jess.com), The Conversation Prism è una mappa visiva del panorama dei social media. È uno studio continuo in etnografia digitale che tiene traccia dei social network dominanti e promettenti e li organizza in base a come vengono utilizzati nella vita di tutti i giorni.

I social media hanno influenzato la società e spostato le strutture di influenza.

2 L'acquisizione di contenuti web in ambito forense

Così come si è avuto modo di introdurre nel capitolo precedente, il quotidiano vivere di ogni cittadino è sempre più immerso nella mediasfera. Questo aspetto non sarebbe di per se un problema se fosse vissuto con la dovuta cognizione, ma la triste realtà mostra invece una mancanza di consapevolezza diffusa, nella quale ogni singolo individuo si pone in situazioni di non sempre facile gestione comportamentale, psicologica, legale o tecnica, che porta di conseguenza il singolo individuo a non sapere valutare il reale impatto delle proprie azioni o viceversa non sapere come difendersi dagli abusi e a volte non essere neppure in grado di riconoscerli

In una società in cui gli strumenti tecnologici rappresentano un elemento imprescindibile del vivere quotidiano, la naturale propensione dell'uomo di rapportarsi al mondo reale con l'uso dei cinque sensi sembrerebbe perdere progressivamente consistenza. Risulta pertanto agevole comprendere non solo come la maggioranza dei reati venga perpetrata, direttamente o indirettamente, a mezzo di apparati digitali, ma come gli stessi siano presenti, a vario titolo, nella quasi totalità delle "*scenae criminum*". [GL2012] In tale contesto la disciplina della Digital Forensics, intesa come quella branca delle scienze forensi attinente alle prove digitali, non può che affermare il proprio crescente ruolo di ausilio del giudice penale in sede di formazione della prova.

In Italia i crimini informatici sono oggetto di studio fin dagli anni '70, ma solo con la convenzione di Budapest del 23 novembre 2001 si sono gettate le basi per un reale contrasto di questa nuova forma di criminalità. In Italia però viene autorizzata la rettificata solo nel 2008 con la legge 18.3.2008, n.48. (n.48/2008) [CCCD2016]

Una prima novità introdotta dalla legge n.48/2008 è rappresentata dalle nuove fattispecie incriminatrici, tutte riguardanti gli strumenti e i dati informatici sotto il profilo del danneggiamento o della frode. È bene a tal punto sottolineare l'impossibilità di ricondurre ad un'unica categoria i reati informatici con riferimento al bene giuridico tutelato. Infatti, pur individuando nel sistema informatico il bene giuridico che il legislatore intende tutelare, resta difficile dare del "reato informatico" una definizione univoca, atteso che vi sono reati direttamente correlati all'uso del computer aventi come scopo la realizzazione da parte dell'autore di un profitto, reati diretti contro il computer aventi come scopo il danneggiamento del sistema informatico, dei programmi e dei dati, o reati correlati all'uso del computer ma diretti a provocare danni a persone e cose. [CCCD2016]

2.1 Crimine informatico

Lo sviluppo delle reti informatiche e telematiche hanno prodotto grandi cambiamenti nelle dinamiche dei rapporti umani, non solo a livello tecnologico ma anche sul piano sociale e culturale. Tale evoluzione, che pur ha portato enormi vantaggi sotto molteplici profili ha inciso anche sul piano giuridico, come su ogni aspetto della nostra società, ed ha creato un terreno fertile per la nascita di nuove forme di reato: i cosiddetti *computer/cyber crimes*.

Un crimine informatico è un fenomeno criminale che si caratterizza nell'abuso della tecnologia informatica sia hardware che software, per la commissione di uno o più crimini. Reati informatici propriamente detti ex L. n°547/93, L. n°48/08 [CM2012]

I reati introdotti dalla legge n. 547/ 93 hanno come caratteristica di avere quale oggetto materiale della condotta quello "informatico"; in altre parole l'azione delittuosa colpisce i sistemi informatici e telematici, i programmi, i dati, le informazioni in essi memorizzati, i documenti informatici, ovvero le comunicazioni informatiche o telematiche. Si tratta di nozioni relativamente nuove per l'ordinamento giuridico, per definire le quali è necessario richiamare il campo tecnico specifico di riferimento, dunque fare ricorso ad ambito extra-penale. Tuttavia, anche nelle discipline tecniche non è facile trovare delle definizioni univoche ed esaustive. Il legislatore, come è accaduto per altre fattispecie di reato, ha preferito non fornire definizioni di tali concetti; ha invece utilizzato una denominazione che fosse onnicomprensiva di tutti gli apparati tecnologici esistenti e che allo stesso tempo potesse adattarsi ai sempre nuovi strumenti

concepiti dall'inarrestabile progresso tecnologico. Così ha introdotto nell'impianto normativo concetti dotati di generalità e astrattezza sufficienti a consentire l'applicazione della norma ad una realtà in continua evoluzione. Ciò premesso, la dottrina ha cercato di riempire di contenuto tali definizioni. Così, tra le tante si segnalano le seguenti:

Ci sono poi reati non informatici, ma commessi con sistemi informatici che possono quindi essere strumenti di aggressione verso altri beni giuridici tutelati dalle norme penali. [VC2017] Esistono inoltre altri contenuti, che non sono essi stessi l'oggetto del reato, ma che risultano comunque di interesse probatorio per quei reati consumati nel mondo fisico e per i quali si rinvengono tracce o indizi nei sistemi informatici. [AG2011]

Il fatto che le tecnologie possano influire sui propositi delittuosi pone il problema della valutazione della loro incidenza sulla condotta costituente un reato, nel senso che possono agevolare la commissione di reati comuni con modalità nuove e mezzi diversi. Nel limbo tra realtà virtuale e realtà materiale, si consumano oggi condotte agevolate dall'erronea, ma molto diffusa, convinzione che attraverso la rete Internet si possa mantenere l'anonimato, che non si venga rintracciati in quanto invisibili, perché celati dietro identità o nomi di comodo, i cosiddetti "nickname". Nell'intimità della propria casa o del proprio ufficio, si può divenire autori o vittime di condotte penalmente rilevanti, consumate nell'immaterialità della "rete", ma con riflessi nella vita reale. L'estrema potenzialità diffusiva delle comunicazioni attraverso Internet, e la convinzione di potersi celare dietro nickname, o identità di comodo, fa sì che creino nuove modalità di aggressione dei beni giuridici tutelati dal nostro ordinamento. [VC2017] La giurisprudenza applica dunque le norme previste a sanzione delle condotte penalmente rilevanti delineate codice, anche quando siano commesse con l'utilizzo di Internet.

Per reati informatici si intendono le condotte criminali compiute tanto su strumenti informatici che per mezzo di essi. Nella letteratura scientifica vengono distinti in computer crimes in senso stretto e cybercrimes. Al proposito si rileva che «*con riferimento a tale tipologia di reati risultano di particolare interesse alcune classificazioni; si possono, infatti distinguere i reati perpetrati per mezzo di sistemi informatici e telematici dai reati realizzati contro i medesimi sistemi, non più intesi come strumenti per compiere atti illeciti ma come oggetti materiali di questi ultimi; ed ancora si può operare una distinzione tra i reati commessi su Internet (o reati*

informatici - telematici propri) dai reati commessi mediante Internet (o reati informatici - telematici impropri), con la precisazione che in quest'ultima categoria rientra un insieme eterogeneo di reati comuni previsti dal codice penale e da alcune leggi speciali» [SB2014]

Osserva una dottrina: «*il ciberspazio rappresenta, da un lato, un nuovo ambiente immateriale al cui interno vengono posti in essere comportamenti penalmente rilevanti di impronta tradizionale e, dall'altro, presenta aspetti strutturali che si traducono in inedite attitudini offensive di beni giuridici classici e già noti alla sistematica del codice penale. Sul piano criminologico i reati commessi a mezzo di Internet si connotano per l'incidenza dell'anonimato nella diffusione degli illeciti e per la indeterminatezza del vittima quale fattore incentivante la scelta delinquenziale*» [RF2001]

Al di là della definizione normativa di reato informatico, un efficace contrasto a tali tipi di reati non può prescindere da una attenta analisi della evoluzione tecnologica che caratterizza l'intero settore e che rende altrettanto mutevoli e sfuggenti le attività illecite.

2.1.1 Prove di reato provenienti dal Web

In generale si preferirà quindi parlare di "prova informatica di un reato" per così racchiudere in una sola definizione le tre categorie appena esposte, ma in particolare l'oggetto di interesse del presente lavoro saranno le "prove di reato provenienti dal Web", per le quali si fornirà di seguito una panoramica che possa delineare al meglio l'area di azione dello strumento informatico studiato e sviluppato nella presente tesi, finalizzato proprio all'acquisizione forense di quei contenuti Web appena menzionati.

A fronte di quanto appena esposto sono stati quindi esclusi volutamente tutte le altre "sorgenti informatiche" che non trovano diretta applicazione nel compito del presente strumento, quali ad esempio: files-sharing, istant messaging, email, servizi di storing in cloud, dark net, etc..

Delitti contro le persone

Tutela dell'onore

Nei reati contro l'onore, il dolo ha natura generica e può assumere anche la forma del dolo eventuale, essendo sufficiente che l'agente faccia consapevolmente uso di espressioni idonee ad assumere portata offensiva. Tale idoneità comprende

necessariamente l'attitudine a raggiungere la sensibilità del soggetto passivo, la quale implica a sua volta la concreta possibilità che quest'ultimo si percepisca come destinatario delle espressioni offensive [CP15060]. Non ogni espressione "forte" o pungente che crei disappunto è automaticamente offensiva ai fini della responsabilità penale nei delitti contro l'onore in quanto la sussistenza di un reato non può essere ancorata alla sensibilità della (presunta) parte offesa, ma rileva, oltre al dolo generico dell'agente, la obiettiva capacità offensiva da giudicarsi in base al significato socialmente condiviso delle parole ed espressioni utilizzate [GL2012].

Nell'ambito del delitto di **ingiuria** (ex art. 594 c.p.), la presenza dell'elemento costituivo del reato dovrà essere ricostruita in base al contesto in cui la lesione dell'onorabilità avvenga mediante le tecnologie informatiche, il che comporta, di conseguenza, delicati problemi di interpretazione al concetto di **diffamazione**.

Ci sarà **ingiuria** quando la comunicazione lesiva sia stata inviata in maniera diretta, ad esempio un messaggio di posta elettronica, una conversazione di istant messaging (IM) e così via. Se invece tale comunicazione entra nella sfera di conoscibilità di più soggetti ad esempio la pubblicazione su un sito Web, su di un forum, o un post effettuato su un social network, risulterà integrato il delitto di **diffamazione.**

Del resto, l'elemento psicologico della diffamazione consiste non solo nella consapevolezza di pronunciare o di scrivere una frase lesiva dell'altrui reputazione, ma anche nella volontà che la frase denigratoria venga a conoscenza di più persone. [GL2012].

Un interessante riflessione ha evidenziato che per l'accertamento della sussistenza del delitto di diffamazione tramite la pubblicazione di espressioni offensive della reputazione altrui sul Web, occorre valutare il contesto in cui tali messaggi appaiano: se ne deduce che non sussiste condotta diffamatoria allorché le predette espressioni siano contenute all'interno di messaggi inviati dai partecipanti ad un newsgroup, qualora da una parte l'utilizzo di soprannomi («alias» o «nicknames») prescelti da ciascuno di essi renda l'individuazione del destinatario delle presunte offese solo potenzialmente individuabile e dall'altra il tenore generale dei contenuti inviati risulti tale da poter ritenere che un utilizzatore di media levatura intellettuale, una volta avuto accesso anche casuale al sito, non potrebbe che percepire l'innocua frivolezza dei messaggi e delle diatribe, anche accese, in essi contenute [TR2007].

Con particolare riferimento all'utilizzo dei social network, il Tribunale di Monza [TM2010] ha ritenuto che in caso di messaggio dal contenuto ingiurioso, inviato tramite un social network da un utente al medesimo registrato e riferibile ad una persona non espressamente citata, ma identificabile con altro utente appartenente al gruppo dei suoi "amici" su quel network, le affermazioni lesive, ove non si possa configurare un "furto di identità", devono ritenersi provenienti dal soggetto a cui nome era stata effettuata la registrazione, il quale è obbligato a risarcire il danno morale subito dalla persona offesa. [TM2010].

Risulta inoltre interessante al progetto in esame, in quanto di difficile cattura con strumenti tradizionali, l'interrogativo che si è posta la giurisprudenza di merito [TM2011] sull'eventuale portata diffamatoria dell'associazione al nome di una persona delle parole "truffa" e "truffatore", operata dal motore di ricerca attraverso il servizio web search denominato Suggest/Autocomplete. Secondo il Tribunale di Milano, chiamato a pronunciarsi in sede civile ex art. 700 c.p.c.: *"l'utente che legge tale abbinamento è indotto immediatamente a dubitare dell'integrità morale del soggetto il cui nome appare associato a tali parole ed a sospettare una condotta non lecita da parte dello stesso. Né appare idonea a svuotare l'abbinamento in oggetto del ritenuto contenuto lesivo, la circostanza che i risultati di ricerca correlati ai due suggerimenti di ricerca di cui si tratta — una volta attivata la ricerca stessa — siano obiettivamente del tutto privi di contenuti offensivi. Il software che consente l'accesso al servizio "Suggest / Autocomplete" costituisce unicamente un'agevolazione offerta da Google ai suoi utenti, la cui eventuale modifica e/ o eliminazione non comprimerebbe in alcun modo la libertà degli stessi di accedere alle ricerche offerte dal motore di ricerca Google — alla stessa maniera di quanto accade per gli altri motori di ricerca. Per tale ragione è il risultato improprio ottenuto con l'applicazione di detto sistema a determinare la responsabilità di chi dello stesso si avvale — irrilevante essendo, in tale prospettiva, l'assenza di ogni intenzionalità lesiva nel provider che lo utilizza. La ritenuta valenza diffamatoria dell'associazione di parole che riguarda il reclamato è innegabilmente di per sé foriera di danni al suo onore, alla sua persona ed alla sua professionalità. La potenzialità lesiva della condotta addebitata alla reclamante — suscettibile, per la sua peculiare natura e per le modalità con cui viene realizzata, di ingravescenza con il passare del tempo stante la notoria frequenza e diffusione dell'impiego del motore di ricerca Google — giustifica il legittimo accoglimento, da parte del giudice di prime cure, del ricorso in via d'urgenza pure sotto il profilo del "periculum in mora"; anche in considerazione della difficoltà obiettiva di provare e*

quindi liquidare il danno nella sua effettiva consistenza, avuto riguardo altresì alla circostanza che il reclamato utilizza il web per la propria attività professionale ". [TM2011]

Di particolare interesse, perché a fronte di tali situazioni i software classici di "acquisizione" non sarebbero in grado di cogliere la sfumatura indicata dalla precedente sentenza, obbligando quindi i consulenti tecnici ad avvalersi di soluzioni non sempre ortodosse e rispettose della "buona pratica" di acquisizione della prova informatica.

Diritto all'immagine

Anche il diritto all'immagine, pur non essendo espressamente previsto in alcuna norma costituzionale, è riconducibile ai diritti fondamentali della personalità di cui all'art. 2 Cost.; tuttavia questo diritto trova ulteriore tutela dal combinato disposto degli artt. 10 c.c. e 96 e 97, L. 22/04/1941, n. 633, sul diritto d'autore.

Quando si parla di diritto all'immagine è possibile riferirsi a diversi aspetti meritevoli di tutela, infatti l'immagine può essere considerata tanto come rappresentazione visiva delle sembianze della persona o riproduzione grafica delle sue fattezze, quanto come dato personale. Elemento fondamentale, che collega entrambe le definizioni, è la riconoscibilità o l'identificabilità.

Il problema sino a qualche anno fa investiva in maniera prevalente i cosiddetti personaggi celebri, ed i rischi derivavano prevalentemente dall'indebito utilizzo di immagini mediante pubblicazione su stampa o trasmissione televisiva. Nel Web siamo tutti minacciati da un rischio di "esposizione mediatica" pur non essendone sempre consapevoli. Le nostre immagini possono essere esposte sui Social Network così come su altri servizi, come per esempio "Google Street View", che ha sollevato in passato problemi a privati cittadini i quali si sono visti ritratti per strada o nella propria abitazione a loro insaputa (e senza darne l'autorizzazione). Non solo dunque un problema di corretto trattamento di dati personali, bensì un intreccio tra privacy, diritto all'immagine e reputazione personale. [MI2016]

La sostituzione di persona su Internet

Accade molto di frequente che un soggetto possa in *chat* "spacciarsi" per un'altra persona, perché sostituisce la propria all'altrui persona, o si attribuisce un falso nome, o un falso stato o una falsa qualità. In chat come nella vita reale tale condotta è reato se induce l'interlocutore in errore.

Ad esempio, la creazione di un account di posta elettronica con un nominativo diverso dal proprio può configurare il reato di sostituzione di persona purché il gestore, o gli utenti, del sito siano tratti in inganno credendo erroneamente di interloquire con una determinata persona mentre si trovano ad avere a che fare con una persona diversa.

In una recente sentenza, la Cassazione, chiamata ad occuparsi di un caso in cui era stato divulgato un numero di cellulare altrui come se fosse il proprio, così ingannando l'interlocutore sulla propria identità, ha insegnato che «*integra il delitto di sostituzione di persona la condotta di chi inserisca nel sito di una "chat line" a tema erotico il recapito telefonico di altra persona associato ad un "nickname" di fantasia, qualora abbia agito al fine di arrecare danno alla medesima, giacché in tal modo gli utilizzatori del servizio vengono tratti in inganno sulla disponibilità della persona associata allo pseudonimo a ricevere comunicazioni a sfondo sessuale*» (Cass., Sez. V, 29 aprile 2013, n. 18826, www.giurisprudenzapenale.com/ 2013/ 06/ 22/ sostituzione-di-persona).

La Cassazione sostiene, cioè, che si perfeziona il reato di cui all'art. 494 c.p. non solo quando ci si attribuisce un nome di battesimo falso, ma anche quando siano falsi i "contrassegni di identità". Ed in tali contrassegni vanno ricompresi, secondo la Suprema corte anche i cosiddetti "*nicknames*" (cioè i soprannomi), utilizzati nelle comunicazioni via Internet che attribuiscono una identità sicuramente virtuale – in quanto destinata a valere nello spazio telematico del web; tuttavia tale identità, seppure virtuale, ha una dimensione concreta in quanto proprio attraverso i nicknames possono avvenire comunicazioni in rete idonee a produrre effetti reali nella sfera giuridica altrui, cioè di coloro ai quali il nickname è attribuito. Nel caso oggetto della decisione da ultimo citata era accaduto che una donna avesse divulgato in una chat di incontri il numero di telefono cellulare della sua ex datrice di lavoro con la quale aveva in corso una causa civile; la vittima, ovviamente del tutto ignara, si era trovata all'improvviso a ricevere, anche in ore notturne, continue telefonate ed sms di persone interessate ad incontri erotici, alcune delle quali non solo l'avevano insultata, ma le avevano anche inviato degli mms con immagini pornografiche.

Sulla stessa linea si sono mosse alle pronunce. Si è così affermato che: «*integra il reato di sostituzione di persona la condotta di colui che crei ed utilizzi un account di posta elettronica, attribuendosi falsamente le generalità di un diverso soggetto, inducendo in errore gli utenti della rete e con il fine di arrecare danno al soggetto le cui generalità siano state abusivamente spese*» (Cass, Sez. V, 28 novembre 2011, n. 18826);

«*il reato di sostituzione di persona (ex art. 494 c.p.) è integrato anche dalla condotta di chi, al fine di procurarsi un vantaggio o arrecare un danno ad altri, utilizza dati anagrafici di terzi (veri ed esistenti) per aprire a loro nome un account e una casella di posta elettronica per iscriversi, nel caso di specie, ad un sito di aste on-line, anche se vi partecipa con un nome di fantasia*» (Cass., Sez. III, 3 aprile 2012, n. 12479).

Allo stesso modo, è reato la condotta di chi crei sul social network facebook un falso profilo (fake), utilizzando nome e foto di un ignaro soggetto, e interagisca con gli "amici" simulando di essere quella persona o utilizzando la sua identità virtuale per offendere la dignità altrui con "post" dal contenuto diffamatorio.

Molestia e disturbo delle persone

La nozione di "molestia" penalmente rilevante ai sensi dell'art. 660 c.p. è stata intesa con notevole ampiezza dalla giurisprudenza di legittimità, la quale ha più volte evidenziato che "il reato di cui all'art. 660 c.p. consiste in qualsiasi condotta oggettivamente idonea a molestare o disturbare terze persone, interferendo nell'altrui vita privata e nell'altrui vita di relazione"; i repertori di giurisprudenza raccontano uno spaccato di vita fatto di condotte di pedinamento, invio di sms, telefonate mute, attese sotto casa della vittima *et similia*.

In materia di molestia o disturbo alle persone, l'art. 660 c.p. è finalizzato a punire quei comportamenti astrattamente idonei a suscitare nella persona offesa e, a volte, anche nella gente, reazioni violente o moti di disgusto o ribellione, i quali influiscono negativamente sul bene giuridico tutelato (l'ordine pubblico); il reato in oggetto non richiede un'estrinsecazione particolare e specifica, ma può concretizzarsi in ogni qualsivoglia condotta oggettivamente idonea a molestare e disturbare terze persone, interferendo nell'altrui vita privata e nell'altrui vita di relazione.

Da ultimo, la Suprema Corte (Sez. I, n. 36/ 2010) ha ritenuto ad esempio che anche una sola telefonata effettuata dopo la mezzanotte sia da considerarsi alla stregua di «molestia» ed integri, pertanto, il reato di cui all'art. 660 c.p.

Al limite del perimetro tratteggiato dal canone di tassatività si pone conclusivamente una recente decisione della Suprema Corte, che affronta la tematica dell'integrazione del reato ex art. 660 c.p. nell'ipotesi di inserimento del numero del telefono cellulare di una persona, all'insaputa di quest'ultima, in un sito internet dedito allo scambio di informazioni di carattere sessuale. Muovendo da un'interpretazione estensiva della

nozione “a mezzo del telefono”, la Corte ha infatti evidenziato che la norma in discorso richiede, per la sua integrazione, che le molestie ed il disturbo siano arrecati alla persona offesa mediante l'utilizzo del telefono; il che è ben ravvisarle nel caso deciso, sebbene nessuna telefonata diretta sia stata effettuata dall'imputata nei confronti della persona offesa, la quale è stata contattata e molestata da terzi, i quali avevano appreso la sua utenza cellulare da un sito internet, nel quale la ricorrente aveva consapevolmente immesso detta utenza. Tanto premesso in punto di fatto, soggiunge la Corte che la persona offesa sia da ritenere il tramite delle successive telefonate moleste, di cui è rimasta vittima la persona offesa, con conseguente ravvisabilità a carico della ricorrente di un volontario concorso nelle molestie arrecate, ben potendo essere la medesima ritenuta quale autrice mediata di dette molestie telefoniche. [GL2012]

Il reato di molestie può essere anche perpetrato a mezzo social network come ha chiarirto la decisione della Suprema Corte (Cass. pen. Sez. I, 11-07-2014, n. 37596) che integra il reato di cui all'art. 660 c.p. l'invio di messaggi molesti, "postati" sulla pagina pubblica di Facebook della persona offesa, trattandosi di luogo virtuale aperto all'accesso di chiunque utilizzi la rete e quindi di “luogo aperto al pubblico”.

La Cassazione ha affrontato un caso di molestie ripetute e frequenti perpetrate ai danni di una giornalista televisiva “presa di mira” dal molestatore, circa le caratteristiche del fisico ed il modo di vestire, sia presso la redazione, sede del suo ufficio, sia con apprezzamenti mediante uso di facebook. [AL2014]

La sentenza in oggetto pone una delle molte questioni applicative in materia di violazione del precetto penale in relazione a condotte realizzate sul web (in particolare facebook). Più precisamente la vicenda attiene alla configurabilità, in tali ipotesi, del reato di molestie, ex art. 660 codice penale, alla luce del tenore letterale della disposizione in esame: “in un luogo pubblico o aperto al pubblico, ovvero col mezzo del telefono”.

Cyberstalking

Occorre precisare che non esiste nel codice penale o in altra fonte giuridica una definizione vera e propria di *cyberstalking*, ma se ne può trovare traccia in alcune recenti sentenze. Con tale termine si intende l’utilizzo di dispositivi informatici di comunicazione come internet di molestare una persona.

Per inquadrare la fattispecie giuridica del reato in esame occorre procedere ad un *excursus* tra i reati di molestia (art.660 c.p.), di minaccia (art.612 c.p.) e di atti persecutori (art.612-*bis* c.p.). [PS2016]

Per integrare la fattispecie del reato di stalking, secondo la dottrina pacifica e l'insegnamento della Corte di Cassazione, è sufficiente la ricorrenza anche di uno solo dei tre effetti della condotta criminosa indicati nella norma, che il legislatore costruisce in maniera alternativa: "*Salvo che il fatto costituisca più grave reato, chiunque reiteratamente,* ***con qualunque mezzo, minaccia o molesta taluno in modo tale da infliggergli un grave disagio psichico*** *ovvero da determinare un giustificato timore per la sicurezza personale propria o di una persona vicina o comunque da pregiudicare in maniera rilevante il suo modo di vivere, è punito, a querela della persona offesa, con la reclusione da sei mesi a quattro anni*". [Art .612-bis CP 2009]

Peraltro, secondo la Suprema Corte, anche due soli episodi di minaccia o molestia possono valere ad integrare il reato di atti persecutori, se abbiano indotto un perdurante stato di ansia o di paura nella vittima [CP2010], trattandosi di reato che deve necessariamente articolarsi in numerose condotte lungo un apprezzabile arco temporale.

Tale criterio minimo individuato dai giudici di legittimità ha orientato anche il variegato panorama della giurisprudenza di merito, con particolare riguardo all'esecuzione delle condotte moleste mediante le tecnologie informatiche (la cui semplicità di utilizzo, unitamente alla sensazione di impunità avvertita da colui che agisce mediante uno schermo e non a viso, rende statisticamente frequente la commissione del delitto di stalking per tale via).

Si è pertanto ritenuto che integrano tale delitto anche due sole condotte di minaccia o di molestia, come tali idonee a costituire la reiterazione richiesta dalla norma incriminatrice, piuttosto che il reiterato invio alla persona offesa di sms e di messaggi di posta elettronica o postali sui cosiddetti social network (ad esempio Facebook) sia come messaggi privati che come post inviati sul profilo della vittima, nonché la divulgazione a mezzo Web di filmati ritraenti rapporti sessuali intrattenuti dall'autore del reato con la medesima. [GL2012] , così come ci ricordano i tristemente noti fatti di cronaca che hanno visto come vittima di suicidio Tiziana Cantone a causa della diffusione sul Web di video rittraenti la medesima durante rapporti sessuali.

Nel ripercorrere i comportamenti molesti addebitati all'indagato, con siffatta pronuncia la Corte di Cassazione pone le fondamenta del fenomeno del ***cyberstalking***, che si

caratterizza per elementi suoi propri tanto da costituire una variante tecnologicamente evoluta del delitto di stalking, che spesso si salda con le tradizionali condotte moleste (appostamenti, telefonate, minacce, pedinamenti et similia) in un unicum inscindibile che opprime la vittima e ne annulla ogni capacità di reazione. [GL2012]

È poi interessante sottolineare anche che il Tribunale di Termini Imerese, con sentenza del 9 febbraio 2012, in merito al delitto di cui all'art. 612-bis c.p ha chiarito ancor meglio che "integrano l'elemento materiale del delitto di atti persecutori"le condotte riconducibili alle categorie del c.d. stalking vigilante (controllo sulla vita quotidiana della vittima), del c.d. stalking comunicativo (consistente in contatti per via epistolare o telefonica, sms, scritte sui muri, ed altri messaggi in luoghi frequentati dalla persona offesa) e del c.d. cyberstalking (costituito dall'uso di tutte quelle tecniche di intrusione molesta nella vita della vittima rese possibili dalle moderne tecnologie informatiche e, segnatamente, dai social network)" ma anche più in generale: forum, community digitali, siti web, etc. [GL2012]

A tal proposito risulta quindi immediatamente chiaro il ruolo fondamentale che assumerebbe uno strumento in grado di "congelare" con indiscutibile certezza, in tempo e luogo (seppure nel Web), la prova informatica capace di esibire le dovute evidenze.

Delitti di pedopornografia

La legge n. 38/2006 ("Disposizioni in materia di lotta contro lo sfruttamento sessuale dei bambini e la pedopornografia anche a mezzo Internet") ha completamente innovato il panorama codicistico dei reati di (pedo)pornografia, conferendo il giusto risalto alla necessità di reprimere le condotte perpetrate mediante lo strumento informatico (oggettivamente agevolate dall'estrema semplicità e velocità di diffusione del dato rilevante - per lo più riproduzioni video-fotografiche - e dalla sensazione di impunità che più o meno intensamente pervade colui il quale agisce per il tramite di uno schermo).

Tanto premesso, in assenza di una puntuale definizione codicistica, invero complessa allorché si fa riferimento a concetti giuridici indeterminati, per "materiale (pedo)pornografico", deve intendersi, in aderenza alla corrispondente nozione fornita dall'art. 1 della decisione quadro del Consiglio europeo 2004/68/GAI del 22 dicembre 2003 [DQ2004], il materiale che ritrae o rappresenta visivamente un minore degli anni diciotto implicato o coinvolto in una condotta sessualmente esplicita, quale può essere anche la semplice esibizione lasciva dei genitali o della regione pubica. Nel panorama

della giurisprudenza di merito, si è ritenuto costituisca materiale pedopornografico ogni rappresentazione finalizzata ad eccitare la sessualità individuale, nel senso che la natura pornografica della rappresentazione di minori in pose che lasciano scoperti integralmente o parzialmente gli organi sessuali, al fine di distinguerla dal materiale di natura diversa (pubblicazioni pubblicitarie, reportages giornalistici, ecc.), deve essere individuata in base all'accertamento della destinazione della rappresentazione ad eccitare la sessualità altrui e dalla idoneità a tale scopo, con la conseguenza che assume rilevanza la natura erotica delle pose e dei movimenti del minore.

Paradigma di questo microcosmo sanzionatorio è l'art. 600-ter c.p. (Pornografia minorile):

"*Alla stessa pena soggiace chi fa commercio del materiale pornografico di cui al primo comma.*

Chiunque, al di fuori delle ipotesi di cui al primo e al secondo comma, con qualsiasi mezzo, anche per via telematica, distribuisce, divulga, diffonde o pubblicizza il materiale pornografico di cui al primo comma, ovvero distribuisce o divulga notizie o informazioni finalizzate all'adescamento o allo sfruttamento sessuale di minori degli anni diciotto, è punito con la reclusione da uno a cinque anni e con la multa da 2.582 euro a 51.645 euro.

Chiunque al di fuori delle ipotesi di cui ai commi primo, secondo e terzo, offre o cede ad altri, anche a titolo gratuito, il materiale pornografico di cui al primo comma, è punito con la reclusione fino a tre anni e con la multa da euro 1.549 a euro 5.164."

L'incriminazione dell'attività di commercio del materiale prodotto tramite lo sfruttamento sessuale dei minori mira a reprimere il fenomeno dello sfruttamento, che si perpetua e si alimenta soltanto se esiste un mercato che assorbe il turpe prodotto: il legislatore ha quindi inteso perseguire l'obiettivo di eliminazione dei canali di distribuzione del materiale, allo scopo di indebolire l'attività diretta alla sua produzione.

Il reato di pornografia minorile, ove commesso per via telematica, è integrato dall'immissione in rete del materiale pedopornografico, in quanto si tratta di condotta idonea a rendere concretamente possibile la diffusione del materiale, attesa la possibilità di accesso al medesimo da parte di un numero indeterminato di persone .

Il delitto di cui al terzo comma della norma in commento, pur prescindendo dall'inciso "anche per via telematica" (la cui utilità semantica risulta invero superflua, alla luce dell'indicazione "con qualsiasi mezzo" di cui alla parte immediatamente precedente

della norma), risulta di ovvio e maggiore interesse nel complesso dell'analisi dei **delitti informatici**.

La giurisprudenza di merito ha opinato che non è sufficiente ad integrare il reato di detenzione di materiale pedopornografico la condotta di chi vi entri in contatto mediante navigazione casuale via internet, ma è necessario che se ne appropri salvando le immagini sul disco fisso del PC o su altri supporti che ne consentano la visione e la riproduzione. Detto salvataggio deve essere consapevole e volontario, dovendosi escludere la penale responsabilità nei casi in cui il materiale rinvenuto sul PC costituisca la mera traccia di una trascorsa consultazione del web, creata dai sistemi di salvataggio automatico del personal computer.

La configurabilità della circostanza aggravante della "ingente quantità" nel delitto di detenzione di materiale pedopornografico impone al giudice di tener conto non solo del numero dei supporti detenuti, dato di per sé indiziante, ma anche del numero di immagini, da considerare come obiettiva unità di misura, che ciascuno di essi contiene.

Cyber bullismo

Nella Gazzetta del 3 giugno 2017 è stata pubblicata la Legge 29 maggio 2017 n. 71 recante "Disposizioni a tutela dei minori per la prevenzione ed il contrasto del fenomeno del cyberbullismo".

Le principali novità introdotte dal provvedimento sono:

- **Definizione di «cyberbullismo»**: con questa espressione si intende "qualunque forma di pressione, aggressione, molestia, ricatto, ingiuria, denigrazione, diffamazione, furto d'identità, alterazione, acquisizione illecita, manipolazione, trattamento illecito di dati personali in danno di minorenni, realizzata per via telematica, nonché la diffusione di contenuti on line aventi ad oggetto anche uno o più componenti della famiglia del minore il cui scopo intenzionale e predominante sia quello di isolare un minore o un gruppo di minori ponendo in atto un serio abuso, un attacco dannoso, o la loro messa in ridicolo".
- **Obiettivo della legge**: il provvedimento intende contrastare il fenomeno del cyberbullismo in tutte le sue manifestazioni, con azioni a carattere preventivo e con una strategia di attenzione, tutela ed educazione nei confronti dei minori coinvolti, sia nella posizione di vittime sia in quella di responsabili di illeciti,

assicurando l'attuazione degli interventi senza distinzione di età nell'ambito delle istituzioni scolastiche.

- **Gestore del sito internet**: si intende il prestatore di servizi della società dell'informazione che, sulla rete internet, cura la gestione dei contenuti di un sito in cui si possono riscontrare le condotte di cyberbullismo; non sono considerati gestori gli access provider, i cache provider e i motori di ricerca.
- **Oscuramento del web**: la vittima di cyberbullismo, che abbia compiuto almeno 14 anni, e i genitori o esercenti la responsabilità sul minore, può inoltrare al titolare del trattamento o al gestore del sito internet o del social *media* un'istanza per l'oscuramento, la rimozione o il blocco di qualsiasi altro dato personale del minore, diffuso nella rete internet. Se non si provvede entro 48 ore, l'interessato può rivolgersi al Garante della Privacy che interviene direttamente entro le successive 48 ore.
- **Ammonimento da parte del questore:** è stata estesa al cyberbullismo la procedura di ammonimento prevista in materia di stalking (art. 612-bis c.p.).

  In caso di condotte di ingiuria (art. 594 c.p.), diffamazione (art. 595 c.p.), minaccia (art. 612 c.p.) e trattamento illecito di dati personali (art. 167 del codice della privacy) commessi mediante internet da minori ultraquattordicenni nei confronti di altro minorenne, fino a quando non è proposta querela o non è presentata denuncia è applicabile la procedura di ammonimento da parte del questore. A tal fine il questore convoca il minore, insieme ad almeno un genitore o ad altra persona esercente la responsabilità genitoriale; gli effetti dell'ammonimento cessano al compimento della maggiore età.
- **Piano d'azione e monitoraggio**: presso la Presidenza del Consiglio è istituito un tavolo tecnico con il compito di redigere un piano di azione integrato per contrastare e prevenire il bullismo e realizzare una banca dati per il monitoraggio del fenomeno.

[AL2017]

Istigazione al suicidio

L'istigazione o aiuto al suicidio è un reato previsto dal codice penale italiano tramite l'articolo 580, che recita:

« Chiunque determina altri al suicidio o rafforza l'altrui proposito di suicidio, ovvero ne agevola in qualsiasi modo l'esecuzione, è punito, se il suicidio avviene, con la reclusione da cinque a dodici anni. Se il suicidio non avviene, è punito con la reclusione da uno a cinque anni, sempre che dal tentativo di suicidio derivi una lesione personale grave o gravissima.

Le pene sono aumentate se la persona istigata o eccitata o aiutata si trova in una delle condizioni indicate nei numeri 1) e 2) dell'articolo precedente. Nondimeno, se la persona suddetta è minore degli anni quattordici o comunque è priva della capacità di intendere e di volere, si applicano le disposizioni relative all'omicidio »

L'appena descritta categoria viene menzionata nel presente documento in quanto rilevante come crimine informatico a mezzo Web per via della sua storia passata, ma anche quella presente. Ancor prima dell'avvento dei Social Network come fenomeno di massa è da sempre stato possibile trovare sul Web siti o forum dedicati alla diffusione delle più svariate tecniche e metodologie per agevolare il suicidio. Anche oggi con l'avvento di nuovi canali comunicativi su Internet, siti e forum dedicati a queste macabre tematiche non sono di certo scomparsi, ma con l'insorgere di nuovi strumenti di aggregazione digitale come i Social Network e la loro sempre più pervasiva presenza nelle normali dinamiche quotidiane, si sono via via introdotte nuove strade digitali dove questi contenuti possono trovare ampia fruizione, potendo per esempio trovare nei gruppi social luoghi digitali in grado di creare community dalla triste capacità di generare un coinvolgimento dinamico.

Per menzionare un episodio che ha avuto grande attenzione mediatica, si farà di seguito accenno al caso “Blu Whale” o anche conosciuto come “gioco del suicidio”.

Il caso inizia a diffondersi a Maggio 2016, partendo da quanto raccontato in un articolo pubblicato sul giornale russo Novaya Gazeta, che narra di una serie di suicidi di ragazzini facenti parte del gruppo F57 del social network russo VKontakte, avvenuti tra Novembre 2015 e Aprile 2016 [NG2016]. Una volta esploso il caso in Russia, passare dalle testate scritte in cirillico a quelle di tutto il resto del mondo è stato un passo breve che ha reso “Blue Whale” un fenomeno di massa facendolo diventare un argomento anche (e soprattutto) di chi non addetto ai lavori.

Lo scopo di questo “gioco al suicidio” prevede la messa in atto di prove via via sempre più auto lesioniste, conferite dai partecipanti passivi del “gioco” al soggetto parte in

causa, il quale raggiungerà la morte nel momento in cui porterà a termine una delle richieste che, in crescente pericolosità, risulterà esser mortale. [BBC2017]

"Blu Whale" ha sollevato e solleva ancora oggi non poche perplessità in merito alla sua reale esistenza, relegandosi a metà strada tra leggenda metropolitana (o sarebbe meglio dire "leggenda della Rete") e realtà. Seppure con il suo alone di perplessità ha comunque contribuito a dare una certa sensibilità sull'argomento dei rischi veicolati dalla Rete e in questo caso dell'istigazione al suicidio a mezzo Web, che a prescindere dal caso specifico, pur con diverse modalità e con meno attenzione mediatica, è una realtà preesistente e necessita delle dovute attenzioni [WPBW].

Tornando nel merito dell'autenticità di "Blu Whale", sembrano essere stati confermati dei casi, il più recente il caso di Pachino avvenuto ad Ottobre 2017 intercettato dalla Polizia italiana, ma molti dubbi ancora permangono, sopratutto sulla reale natura dell'episodio: è esso reale manifestazione o "solo" emulazione?

Infatti le prime indagini effettuate dalla polizia di stato ci dicono che il fenomeno è cresciuto perché se ne parla molto, ma le verifiche sulle segnalazioni arrivate alla Polizia Postale italiana mirano a mettere in luce soprattutto i casi di emulazione. [PS2017].

Quindi anche se nella sua forma originale, così come proposto dai *media* russi, il caso "Blue Whale" potrebbe non esistere come fenomeno autentico, potendo quindi essere una delle tante leggenda della Rete, l'argomento in questione risulta comunque di delicata importanza del quale occorre monitorare l'evoluzione e le potenziali emulazioni che, se anche tali, non risulterebbero di certo meno gravi o meno importanti.

Delitti contro il patrimonio

La protezione del diritto d'autore

"*Tutte le opere dell'ingegno di carattere creativo che appartengono alle scienze, alla letteratura, alla musica, alle arti figurative, all'architettura, al teatro e alla cinematografia, qualunque ne sia il modo o l'espressione, formano oggetto del diritto d'autore*" (art. 2575 c.c.)

"*Sono protette ai sensi di questa legge le opere dell'ingegno di carattere creativo che appartengono alla letteratura, alla musica, alle arti figurative, all'architettura, al teatro ed alla cinematografia, qualunque ne sia il modo o la forma di espressione. Sono*

altresì protetti i programmi per elaboratore come opere letterarie ai sensi della convenzione di Berna sulla protezione delle opere letterarie ed artistiche ratificata e resa esecutiva con legge 20 giugno 1978, n. 399, nonché le banche di dati che per la scelta o la disposizione del materiale costituiscono una creazione intellettuale dell'autore." (art. 1 L. 22/1941)

Leggi di riferimento art.

- art 2575 c.c.
- Legge 22 aprile 1941 n. 633 - Protezione del diritto d'autore e di altri diritti connessi al suo esercizio
- L. 248/2000
- 2001/29 - Direttiva sul Diritto d'Autore e diritti connessi nella Società dell'Informazione

La Legge n. 248 del 18 agosto 2000 (http://www.camera.it/parlam/leggi/00248l.htm) ha modificato la legge n. 633 del 22 aprile 1941, denominata "Protezione del diritto d'autore e di altri diritti connessi al suo esercizio", prevedendo ulteriori disposizioni a tutela del diritto d'autore anche al fine di **combattere la contraffazione e la pirateria realizza sul web**

Profili strutturali classici del diritto civile ed un armamentario sanzionatorio penalistico si intrecciano nella complessa normativa a tutela del diritto d'autore, evidenziando l'importanza di una concezione patrimonialistica come punto cardine per l'individuazione del bene giuridico tutelato. [GL2012]

La protezione del diritto d'autore è sicuramente una delle questioni dibattute da più tempo quando si tratta di Internet, utilizzato dai suoi albori come mezzo principale di diffusione di contenuti non autorizzati; ciò ha avuto e continua ad avere una forte attenzione legislativa ed è costantemente argomento che pone nuove frontiere, perciò risulterà arduo, oltre che fuori degli scopi del presente documento, poterne fare una panoramica completa ed esaustiva, ma è altresì importante menzionarlo in quanto sicuramente uno degli aspetti dove trova ampio utilizzo oggi, come nel passato, l'acquisizione dell'evidenza informatica.

Sono sotto gli occhi di tutti i casi relativi a siti web sottoposti a sequestro, non solo quando pubblicavano contenuti coperti da copyright, ma anche quando si limitavano a

pubblicare elenchi di link per raggiungere i contenuti ospitati su altri server o servizio (esempio Torrent). [MI2016]

Protezione del marchio

Una delle preoccupazioni maggiori di chi avvia o ha già un'attività economica è quella del rapporto tra diffusione globale di un sito internet e tutela del marchio.

Come si sa, una volta che si immettono contenuti in internet, questi divengono globalmente accessibili (indipendentemente dal .it, .com, .ue, ecc). Questo vuol dire che il proprio marchio e/o i propri prodotti possono essere raggiungibili (e quindi copiati) in qualsiasi parte del globo.

La contraffazione non produce danni sono per l'economia, ma è anche un danno per i singoli Stati, per le imprese e per i consumatori finali. [CE2016]

Per lo Stato la contraffazione rappresenta un furto: questo fenomeno comporta un'evasione totale, come è emerso dal rapporto CENSIS, con il risultato che il mancato introito ricade sui cittadini ai quali vengono aggiunte ulteriori imposte per coprire il disavanzo che si è creato.

Per la società è un crimine che si compie con lo sfruttamento di lavoro, con riciclaggio di denaro, con la creazione di un'importante area d'investimento per la criminalità organizzata e con un aumento dei casi di corruzione.

Per le imprese è una perdita di qualità, d'immagine, di innovazione, di profitti e di investimenti. Come emerge dall'articolo di Giulia Crivelli, scrittrice per il Sole 24 Ore "*(...) le aziende spendono sempre di più per combattere chi produce e vende prodotti falsi (...): si stima ad esempio che in un anno investa dai 10 ai 20 milioni di dollari in azioni legali legate alla contraffazione.*"

Importanti sono stati gli interventi della Corte di Giustizia Europea, la quale si è espressa su alcuni casi di contraffazione che le sono stati sottoposti. Attraverso tali sentenze sono stati individuati i fenomeni illeciti che possono essere compiuti online, permettendo così al titolare del diritto violato di identificarli e intraprendere così le azioni di tutela regolate dalle normative in materia.

Per spiegare come avviene la violazione del marchio nel web, è importante dare alcune definizioni dei termini di 'contraffazione', 'marchio d'impresa' e 'nomi a dominio', i quali sono strettamente collegati tra di loro e costituiscono i punti fondamentali di questo elaborato.

Secondo i termini dell'accordo *TRIPs – Trade Related aspects of Intellectual Property rights* – negoziato in seno alla World Trade Organization, "*il termine contraffazione è esclusivamente riferibile ai casi di violazione del marchio di fabbrica e di commercio, includendo qualunque azione tesa a realizzare prodotti che imitano le caratteristiche di un altro prodotto con lo scopo di trarre in inganno l'acquirente e di ricavare un illecito guadagno dalla vendita di beni non autentic*i".

Di conseguenza, "*l'espressione "beni con marchio" contraffatto indica qualunque tipo di bene, incluso l'imballaggio, che rechi senza autorizzazione, un marchio di fabbrica identico a quello registrato per lo stesso genere di prodotto (...) e che di conseguenza violi i diritti del legittimo titolare del marchio medesimo*"[CS2006].

Il marchio com'è affermato dall'art 7 del Codice della Proprietà Industriale, D.lgs 30/2005, " *può essere costituito da " (...) tutti i segni suscettibili di essere rappresentati graficamente, in particolare le parole, compresi i nomi di persone, i disegni, le lettere, le cifre, i suoni, la forma del prodotto o della confezione di esso, le combinazioni o le tonalità cromatiche, purché siano atti a distinguere i prodotti o i servizi di un'impresa da quelli di altre imprese*".

Il marchio è dunque il più importante segno distintivo dell'impresa, di cui contraddistingue i prodotti e i servizi. Sul piano economico, oggi è tutelato non solo come strumento per informare il pubblico della provenienza dei prodotti, ma anche come simbolo che il pubblico ricollega a quei prodotti per i quali esso è utilizzato e su cui si concentra il valore di mercato. E' inoltre indicatore dell'origine, e dunque fornisce una garanzia sulle caratteristiche e sulla qualità dei prodotti.

Ipotesi di violazione di Marchio in Internet

- *Usurpazione Classica*: utilizzo non autorizzato di un marchio altrui su un sito internet (non nel dominio) con modalità distintive (e-commerce, banner pubblicitari, pop up)
- *Domaing rabbing*: vienere gistratoun DN confondibile con un marchio noto a scopo di illecito concorrenziale es. www.gogle.it
- *Cybersquatting*: registrazione abusiva di un DN identico o simile ad altro noto a scopo di disturbo
- *Framing*: collegamento ipertestuale con altro sit oche quindi appare nel frame del sito richiamante

- *Deep linking* : collegamento ipertestuale con altro sito senza passare per la home page di quest'ultimo.
- *Metatags*: parole chiavi inserite nel linguaggio HTML che leggono i motori di ricerca per indicizzare i siti. Es. Ditta di abbigliamento che impiega come metatag il marchio di un produttore famoso di abbigliamento. [LM2009]

Una prima fonte normativa per la **tutela del nome a dominio** è il Codice della Proprietà Industriale, il quale lo considera come **un segno distintivo che è collegato all'attività dell'impresa in quanto ne identifica il sito web**. Per tale motivo esso è soggetto alla medesima disciplina e tutela prevista per gli altri segni distintivi dell'impresa. Un primo importante principio è sancito dall'art 22 del CPI, il quale si occupa di stabilire "l'unitarietà dei segni distintivi". L'articolo in esame, infatti, afferma: "*E' vietato adottare come nome a dominio di un sito usato nell'attività economica un segno uguale o simile all'altrui marchio se, a causa dell'identità o dell'affinità tra l'attività d'impresa dei titolari di quei due segni ed i prodotti o servizi per i quali il marchio è adottato, possa determinarsi un rischio di confusione per il pubblico che può consistere anche in un rischio di associazione fra i due segni*"

Oltre al divieto di utilizzare come nome a dominio il marchio altrui per prodotti appartenenti allo stesso settore merceologico, al secondo comma l'articolo fa divieto di utilizzare un marchio rinomato come nome a dominio, anche se i prodotti contraddistinti da tale marchio, non appartengono alla stessa categoria di quelli della società concorrente. L'articolo in esame infatti afferma: "*Il divieto si estende all'adozione come nome a dominio di un sito usato nell'attività economica di un segno uguale o simile ad un marchio registrato per prodotti o servizi anche non affini, che goda dello stato di rinomanza se l'uso del segno senza giusto motivo consente di trarre indebitamente vantaggio del carattere distintivo o della rinomanza del marchio e reca pregiudizio agli stessi*".

L'art. 22 ha dunque un importante obiettivo: evitare che l'utilizzo nel nome a dominio di un marchio altrui possa generare negli utenti del web un rischio di confusione o di associazione tra due aziende concorrenti, conducendo così allo sviamento della clientela e danneggiando l'azienda titolare del marchio violato.

Frode informatica e Phishing

L'art. 10 l. 547/ 93 ha introdotto nell'impianto codicistico l'art. 640 ter c.p. rubricato "**frode informatica**).

Il delitto di frode informatica ripropone la condotta sanzionata dalla truffa e ne costituisce una forma per così dire "aggiornata".

La condotta è sostanzialmente uguale in entrambi i reati consistendo nel "procurare a sé o ad altri un ingiusto profitto con altrui danno" ma differiscono le modalità di realizzazione del vantaggio economico che, mentre nella truffa si sostanziano negli "artifici e raggiri, inducendo taluno in errore", nella frode informatica sono indicate in tassative modalità: "alterando in qualsiasi modo il funzionamento di un sistema informatico o telematico", ovvero "intervenendo senza diritto con qualsiasi modalità su dati, informazioni o programmi contenuti in un sistema informatico o telematico o ad esso pertinenti". [VC2017] Come ha ripetutamente sottolineato, sul punto, la Suprema corte,

«*la novella tracciata ex art. 640 ter c.p. (c.d. Frode informatica) (cfr. L. n. 547 del 1993) ha la medesima struttura e quindi i medesimi elementi costitutivi della truffa dalla quale si differenzia solamente perché l'attività fraudolenta dell'agente investe non la persona (soggetto passivo), di cui difetta l'induzione in errore, bensì il 'sistema informatico' di pertinenza della medesima, attraverso la manipolazione di detto sistema*» (Cass., Sez. VI, 26 febbraio 2009, n. 8755, CED, 243238);

«*per la giurisprudenza di questa Corte, il reato di frode informatica - che postula necessariamente la manipolazione del sistema - presenta la medesima struttura e gli stessi elementi costitutivi della truffa, con l'unica differenza che non viene indotto in errore il la persona del soggetto passivo, ma l'attività fraudolenta dell'agente investe il sistema informatico riferibile al suddetto*» (Cass., Sez.VI, 5 febbraio 2004, n. 4576, GI, 2004, 1881).

La circostanza comunque che la frode informatica rappresenti una "evoluzione" della truffa ha fatto sagacemente sostenere ad una dottrina che «*si può, quindi, affermare che, così come l'autore di truffe ieri, il "truffatore" informatico oggi si distingue per fantasia, intraprendenza intellettuale, spesso anche per intelligenza e competenza tecnica. Proprio per questo è un criminale particolarmente insidioso che spesso valuta con largo anticipo non solo come conseguire un profitto illecito, ma anche come non essere individuato*» [CA2006]

La norma in questione, a differenza della truffa, prevede dunque due diverse condotte, l'alterazione in qualsiasi modo del funzionamento di un sistema informatico o telematico e l'intervento senza diritto con qualsiasi modalità su dati, informazioni o programmi contenuti in un sistema informatico o telematico o ad esso pertinenti, che rappresentano gli artifici o i raggiri propri della frode informatica, in quanto modificano il funzionamento del sistema telematico o informatico per compiere operazioni che portano a risultati non voluti dal suo titolare ma vantaggiosi per l'agente. Al proposito, si è osservato che «*l'alterazione in qualsiasi modo del "funzionamento di un sistema informatico o telematico" consisterebbe in un'alterazione "estrinseca" del sistema, determinata, agendo sul software (con la sostituzione del programma o con modifiche strutturali di quest'ultimo) e/ o hardware. Al contrario "l'intervento senza diritto, con qualsiasi modalità su dati, informazioni o programmi di un sistema" sembrerebbe contemplare modifiche "intrinseche" al sistema operativo, dirette ad alterare, oltre che i dati, gli esiti delle elaborazioni, con inserimento di informazioni, e delle correlazioni logiche del programma*» [CA2006].

L'alterazione può essere ottenuta in due maniere: o agendo sul software, la componente logica del computer, cioè su programmi, dati, informazioni installati e memorizzati in un apparato con capacità di elaborazione; ovvero operando sull'hardware, cioè sulle parti elettroniche, meccaniche, magnetiche, ottiche che ne consentono il funzionamento, in modo tale da far compiere operazioni diverse rispetto a quelle per le quali la macchina è stata programmata, ad esempio con un'azione volta a modificarne la componente fisica. [VC2017] Per sistema informatico deve intendersi, secondo una recente sentenza della Suprema corte, un «*complesso di apparecchiature destinate a compiere una qualsiasi funzione utile all'uomo, attraverso l'uso, anche parziale, di tecnologie informatiche*» (Cass., Sez. II, 15 aprile 2011, n. 17748), ovverosia attraverso l'uso di una tecnologia che sia «*caratterizzata – per mezzo di una attività di "codificazione" e "decodificazione" – dalla "registrazione" o "memorizzazione" di impulsi elettronici, su supporti adeguati di "dati", e cioè da rappresentazioni elementari di un fatto, effettuate attraverso simboli (bit), in combinazioni diverse, e dalla elaborazione automatica di tali dati [... la quale] genera "informazioni", costituite da un insieme più o meno vasto di dati organizzati secondo una logica che consente loro di esprimere un particolare significato per l'utente*» (Cass., Sez. II, 15 aprile 2011, n. 17748,).

Va inoltre evidenziata la natura autonoma del reato di frode informatica rispetto a quello di truffa, del quale quindi non è fatto aggravato, avendo la Cassazione osservato come

«*la fattispecie di cui all'art. 640 ter integri senz'altro un'autonoma figura di reato, a differenza di quanto si è invece ritenuto in giurisprudenza a proposito della ipotesi di truffa aggravata per il conseguimento di erogazioni pubbliche, prevista dall'art. 640-bis cod. pen., ormai pacificamente ricondotta nel novero delle circostanze aggravanti rispetto al reato "base" di truffa ex art 640 cod. pen. (Cass., Sez. un., 26 giugno 2002, P.G. in Proc. Fedi)*» (Cass., Sez. I, 6 maggio 2011, n. 17748, CED, 250113).

È stato altresì riconosciuto il concorso formale tra i delitti di accesso abusivo a un sistema informatico e quello di frode informatica: «*trattasi di reati totalmente diversi, il secondo dei quali postula necessariamente la manipolazione del sistema, elemento costitutivo non necessario per la consumazione del primo: la differenza fra le due ipotesi criminose si ricava, inoltre, dalla diversità dei beni giuridici tutelati, dall'elemento soggettivo e dalla previsione della possibilità di commettere il reato di accesso abusivo solo nei riguardi di sistemi protetti, caratteristica che non ricorre nel reato di frode informatica*» (Cass, Sez. VI, 14 dicembre 1999, n. 3067, CP, 2000, 2990).

Tra le modalità in cui si realizzano i casi di frode informatica, vi è una continuo incremento del c.d. ***phishing***.

Si tratta di un fenomeno, purtroppo diffuso, volto al furto di dati personali al fine di lucro che trae origine dall'invio casuale o mirato di email (ma anche chat o post su social network) che riproducono la grafica e i loghi ufficiali di siti aziendali o istituzionali come quelli postali o bancari.

Tali messaggi, di solito, comunicano all'utente messaggi con informazioni tali da indurlo a cliccare su link associati a pagine Web "esca", inducendolo per esempio all'inserimento di password che autorizzano pagamenti o numeri di carte di credito.

La Cassazione si è occupata di suddetto fenomeno con riferimento al caso di false comunicazioni, che venivano inviate ad ignari soggetti, con le quali si chiedeva la trasmissione di dati personali, da parte di un sito clone di Poste Italiane. In quest'occasione, la Corte, dopo aver individuato la nozione di phishing come «quell'attività illecita in base alla quale, attraverso vari stratagemmi (o attraverso fasulli messaggi di posta elettronica, o attraverso veri e propri programmi informatici e malware) un soggetto riesce ad impossessarsi fraudolentemente dei codici elettronici (user e password) di un utente, codici che, poi, utilizza per frodi informatiche

consistenti, di solito, nell'accedere a conti correnti bancali o postali che vengono rapidamente svuotati», (Cass., Sez. II, 11 marzo 2011, n. 9891, CED, 249675), ha affermato il principio secondo il quale «*integra il reato di frode informatica, e non già soltanto quello di accesso abusivo ad un sistema informatico o telematico, la condotta di introduzione nel sistema informatico delle Poste italiane S.p.A. mediante l'abusiva utilizzazione dei codici di accesso personale di un correntista e di trasferimento fraudolento, in proprio favore, di somme di denaro depositate sul conto corrente del predetto*» (Cass., Sez. II, 11 marzo 2011, n. 9891, CED, 249675).

Lo strumento in essere nel presente documento di tesi avrà come obiettivo quello di raccogliere (anche) tutte le evidenze di quelle pagine (siti, entità social, etc) con finalità fraudolente. Nella fase di acquisizione risulterà importante raccogliere tutte quelle informazioni che vanno "oltre" al contenuto visibile, poiché risulterà fondamentale poter analizzare anche il codice eseguito all'interno del browser dell'utente e come questo abbia determinato l'evoluzione del "*document object model*" (DOM) della pagina. Sarà quindi necessario avere a disposizione sia il codice raw che il tracciamento delle esecuzione avvenute e l'evoluzione del DOM, così da poter analizzare puntualmente ogni singolo evento avvenuto durante la fase di visualizzazione per poter determinare se siano stati eseguiti codici malevoli tali da sfruttare vulnerabilità del browser per compiere attività fraudolente. Ma anche in presenza di codice non così elaborato e quindi in assenza di esecuzione di determinati payload in grado di sfruttare specifici "exploit" sarà comunque altresì importante capire che tipo di eventi si sono verificati e con quali finalità.

Un esempio molto attuale e di particolare interesse, è l'esecuzione di script in grado di effettuare il "*mining*" di **cripto-valute** all'interno del browser, così da sfruttare l'hardware degli utenti. Questo tipo di attività si posiziona in una frontiera grigia ad oggi ancora non delineata dalla giurisprudenza, ma che risulta particolarmente interessante studiare per capirne i risvolti che tali attività possono avere da un punto di vista legale oltre che tecnico e delle relative conseguenze.

A tutti gli effetti non avviene un vero e proprio accesso abusivo alla macchina dell'utente, ma è certa però l'assenza della necessaria comunicazione da parte del sito di quanto sarà compiuto con le risorse hardware e software dell'utente.

Le attività di estrazione di cripto-valute sono operazioni molto "impegnative" che portano l'hardware ad essere sfruttato intensamente con relativo surriscaldamento della

CPU/GPU con relativo rischio di usura e maggiori consumi di energia elettrica, in grado potenzialmente di arrecare danno per l'utente.

Delitti contro la società

Contrasto al terrorismo

Le modifiche in tema di contrasto al terrorismo internazionale intervenute con la legge 17 aprile 2015 n. 43 sono state, in ordine di tempo, l'ultimo banco di prova di questa necessità di soluzioni condivise, idonee a "fronteggiare" le tecnologie. [CCCD2016]

Con la pubblicazione in Gazzetta Ufficiale 20 aprile 2015, n. 91 la Legge 17 aprile 2015, n. 43 recante Conversione in legge, con modificazioni, del decreto-legge 18 febbraio 2015, n. 7, recante misure urgenti per il contrasto del terrorismo, anche di matrice internazionale si viene a delineare la possibilità di applicare la misura della sorveglianza speciale di pubblica sicurezza ai potenziali "foreign fighters"; l'introduzione di una nuova figura di reato destinata a punire chi organizza, finanzia e propaganda viaggi per commettere condotte terroristiche; [AL2015]

Vengono quindi introdotte misure per il contrasto alle **attività di proselitismo attraverso internet**. Quando i reati di terrorismo, l'istigazione e l'apologia del terrorismo sono commessi tramite strumenti informatici e telematici, sono anzitutto previste aggravanti di pena. Si stabilisce poi che la polizia postale e delle comunicazioni debba costantemente tenere aggiornata una black-list dei siti internet (e di conseguenza la monitorizzazione costante di tutte quelle zone della Rete ove è possibile trovare tale propaganda, nella fattispecie: social network - pagine Facebook in primis - siti web dedicati, forum, etc.) che vengano utilizzati per la commissione di reati di terrorismo, anche al fine di favorire lo svolgimento delle indagini della polizia giudiziaria, effettuate anche sotto copertura. [AL2015]

Inoltre, tra le modifiche più importanti al testo durante l'esame in Commissione alla Camera, va ricordato lo stralcio della norma che autorizzava la polizia a entrare all'interno dei computer da remoto per intercettare le comunicazioni via web dei sospettati di terrorismo. [AL2015]

Propaganda di idee razziste

Due leggi sono di particolare interesse:

- Legge n.13 ottobre 1975, n.654,
- Legge 25 giugno 1993, n.205, conosciuta anche come “Legge Mancino”.

Legge n.13 ottobre 1975 n.654, ratifica ed esecuzione della convenzione internazionale sull'eliminazione di tutte le forme di discriminazione razziale, aperta alla firma a New York il 7 marzo 1966 pubblicata nella Gazz. Uff. 23 dicembre 1975, n. 337, S.O.

E' vietata ogni organizzazione, associazione, movimento o gruppo avente tra i propri scopi l'incitamento alla discriminazione o alla violenza per motivi razziali, etnici, nazionali o religiosi. Chi partecipa a tali organizzazioni, associazioni, movimenti o gruppi, o presta assistenza alla loro attività, è punito, per il solo fatto della partecipazione o dell'assistenza, con la reclusione da sei mesi a quattro anni. Coloro che promuovono o dirigono tali organizzazioni, associazioni, movimenti o gruppi sono puniti, per ciò solo, con la reclusione da uno a sei anni. [LM1993]

La legge nulla dice in ordine al mezzo di diffusione della propaganda razzista, permettendo dunque di perseguire la condotta anche attraverso strumenti telematici, quali siti Web, social network, forum, etc. Conferma di questa possibilità è l'operazione di sequestro preventivo e relativo oscuramento di siti Web dal contenuto antisemita effettuato dalla Procura di Bolzano con l'operazione denominata “Operazione Holywar” in data 12 Aprile 2013, dive viene espressamente indicato:

“*Due dei soggetti perquisiti ed altri ancora da individuare sono indagati per aver promosso e diretto un gruppo avente tra i propri scopi l'incitamento alla discriminazione del popolo ebraico per motivi religiosi (delitto di cui all'art.3, comma 3, Legge 13 ottobre 1975, n.654), mentre gli altri sono indagati per aver diffuso idee fondate sull'odio razziale per motivi religiosi nei confronti della Comunità Ebraica (delitto di cui all'art.3 comma 1, lett.a, Legge 13 ottobre 1975, n.654)*”. [PB2013]

La legge 25 giugno 1993 n. 205 è invece una norma della Repubblica Italiana che sanziona e condanna gesti, azioni e slogan legati all'ideologia nazifascista, e aventi per scopo l'incitazione alla violenza e alla discriminazione per motivi razziali, etnici, religiosi o nazionali. La legge punisce anche l'utilizzo di simbologie legate a suddetti movimenti politici.

Emanata con il decreto legge 26 aprile 1993 n. 122 - convertito con modificazioni in legge 25 giugno 1993, n. 205 - è nota come legge Mancino, dal nome dell'allora Ministro dell'Interno che ne fu proponente (il democristiano Nicola Mancino).

Essa è oggi il principale strumento legislativo che l'ordinamento italiano offre per la repressione dei crimini d'odio. [LM1993]

L'intervento legislativo della Legge Mancino si è quindi tradotto in un inasprimento del trattamento sanzionatorio, nonché in un ampliamento dell'ambito di tutela, attraverso l'estensione della rilevanza penale anche alle manifestazioni discriminatorie attinenti alla sfera "religiosa", oltre a quelle razziale, etnica e nazionale, e mediante l'incriminazione di singoli "atti" di contenuto discriminatorio accanto alle condotte di "incitamento" o di provocazione di altri a porre in essere azioni di tale natura.

Come è stato scritto[LP2012], una molteplicità di delitti commissibili in Internet offendono beni giuridici diversi, che sono indisponibili data la loro rilevanza collettiva o addirittura pubblica, per cui sono perseguibili d'ufficio.

L'utilizzo di Internet integra l'ipotesi aggravata di cui all'art. 595, co. 3, c.p. (offesa recata con qualsiasi altro mezzo di pubblicità), poiché la particolare diffusività del mezzo usato per propagare il messaggio denigratorio rende l'agente meritevole di un più severo trattamento penale. [TL2012]

Apologia di fascismo

Nell'ordinamento italiano, l'apologia del fascismo è un reato previsto dall'art. 4 della legge 20 giugno 1952, n. 645 (contenente "Norme di attuazione della XII disposizione transitoria e finale (comma primo) della Costituzione"), anche detta Legge Scelba.

«*[...]quando un'associazione, un movimento o comunque un gruppo di persone non inferiore a cinque persegue finalità antidemocratiche proprie del partito fascista, esaltando, minacciando o usando la violenza quale metodo di lotta politica o propugnando la soppressione delle libertà garantite dalla Costituzione o denigrando la democrazia, le sue istituzioni e i valori della Resistenza, o svolgendo propaganda razzista, ovvero rivolge la sua attività alla esaltazione di esponenti, principi, fatti e metodi propri del predetto partito o compie manifestazioni esteriori di carattere fascista.*»

La legge n. 645/1952 sanziona chiunque promuova od organizzi, sotto qualsiasi forma, la costituzione di un'associazione, di un movimento o di un gruppo avente le

caratteristiche e perseguente le finalità di riorganizzazione del disciolto partito fascista, oppure chiunque pubblicamente esalti esponenti, princìpi, fatti o metodi del fascismo, oppure le sue finalità antidemocratiche.

È vietata perciò la ricostruzione del PNF, del PFR e del NSDAP. Ogni tipo di apologia è punibile con un arresto dai 18 mesi ai 4 anni.

La norma prevede sanzioni detentive anche per i colpevoli del reato di apologia, più severe se il fatto riguarda idee o metodi razzisti o se è commesso con il mezzo della stampa. La pena detentiva è accompagnata dalla pena accessoria dell'interdizione dai pubblici uffici. [L201952]

Nel 2016 sono state poi integrate modifiche alla legge per così poter contemplare anche il mezzo telematico come strumento di divulgazione:

All'articolo 4 della legge 20 giugno 1952, n. 645, sono apportate le seguenti modificazioni:

a) dopo il primo comma, è inserito il seguente:

«Commette apologia di fascismo e soggiace alla pena di cui al primo comma chiunque produce, distribuisce, diffonde o vende, direttamente o ***con qualsiasi modalità, anche telematica****, beni mobili raffiguranti immagini o simboli che si richiamano univocamente all'ideologia fascista o nazifascista ovvero ad associazioni, movimenti o gruppi aventi le caratteristiche e perseguenti le finalità indicate nell'articolo 1»;*

*b) al terzo comma, dopo le parole: «con il mezzo della stampa» sono aggiunte, in fine, le seguenti: «****o mediante la rete internet. In quest'ultimo caso, il pubblico ministero dispone l'ordine di rimozione dei contenuti illeciti, secondo quanto previsto dall'articolo 2, comma 4, del decreto-legge 18 febbraio 2015, n. 7, convertito, con modificazioni, dalla legge 17 aprile 2015, n. 43****».* [SR2016]

Istigazione a delinquere / apologia di reato

L'articolo 414 Codice Penale, recita così:

"*Chiunque* ***pubblicamente*** *istiga a commettere uno o più reati è punito, per il solo fatto dell'istigazione:*

1) con la reclusione da uno a cinque anni, se trattasi di istigazione a commettere delitti;

2) con la reclusione fino a un anno, ovvero con la multa fino a duecentosei euro, se trattasi di istigazione a commettere contravvenzioni.

Se si tratta di istigazione a commettere uno o più delitti e una o più contravvenzioni, si applica la pena stabilita nel numero 1.

Alla pena stabilita nel numero 1 soggiace anche chi **pubblicamente** fa l'apologia di uno o più delitti. La pena prevista dal presente comma nonché dal primo e dal secondo comma **è aumentata se il fatto è commesso attraverso strumenti informatici o telematici**.

Fuori dei casi di cui all'articolo 302, se l'istigazione o l'apologia di cui ai commi precedenti riguarda delitti di terrorismo o crimini contro l'umanità la pena è aumentata della metà. La pena è aumentata fino a due terzi se il fatto è commesso attraverso strumenti informatici o telematici".

Nell'articolo 414, precedentemente menzionato, viene più volte esposta la natura "pubblica" dell'istigazione, la quale è ravvisabile nel caso di documenti diffusi su siti internet liberamente accessibili. Infatti, l'articolo 266, comma 4, del Cp definisce il reato avvenuto "pubblicamente" quando il fatto è commesso "col mezzo della stampa o con altro mezzo di propaganda" ed è evidente che un sito internet a libero accesso ha una potenzialità diffusiva indefinita, tanto da poter essere equiparato alla stampa.

Tale "interpretazione" è confermata dalla Corte di Cassazione, Sezione I, 6 ottobre 2015 - 1° dicembre 2015 n. 47489 - dove nel merito della "*Istigazione a delinquere o a disobbedire e apologia di reato o sovversiva — Associazione terroristica IS — Documento apologetico — **Diffusione in Internet** — Reato (Cod. pen., art. 414)*" afferma che

"*integra il reato di apologia di cui all'art. 414, 3° comma, c.p. la condotta dell'agente **consistita nel diffondere su un sito Internet (privo di vincoli di accesso)** un documento che solleciti l'adesione dei potenziali lettori allo «Stato islamico», esaltandone la natura combattente, la diffusione e la sua espansione anche con l'uso delle armi, posto che tale modalità ha una potenzialità diffusiva indefinita.*"

Quindi quando l'istigazione a delinquere o l'apologia di reato sono commessi su internet scatta l'aggravante con conseguente aumento della pena. Secondo però la Cassazione [CS2017] non è sufficiente che il messaggio sia comunicato in una chat privata; al contrario deve essere leggibile da tutti, come nel caso di un post su Facebook, di un articolo su un sito Web o di un commento ad un articolo apparso su internet da una pagina accessibile a tutti senza restrizioni. Insomma, il messaggio deve possedere connotazioni di potenzialità diffusiva, conseguente al fatto di essere destinato a un

numero indeterminato di soggetti e comunque non riconducibili a un ambito strettamente interpersonale.

Non vi sono quindi dubbi sulle ragioni per le quali qualsiasi contenuto veicolato per mezzo del Web, con qualsiasi tipo di strumento (come i social network, siti, blog, forum, etc.), che connoti una incitazione a delinquere o apologia di reato, sarà di nostro stretto interesse in qualità di evidenza informatica oggetto di acquisizione.

2.2 Acquisizione dell'evidenza informatica e Convenzione di Budapest

I reati informatici sono disciplinati nel codice penale e in alcune leggi speciali successive. La convenzione di Budapest del 23 novembre 2001 del Consiglio d'Europa sulla criminalità informatica (ratificata con la legge 48/ 2008) può essere considerata il primo vero accordo internazionale riguardante i crimini commessi attraverso internet o altre reti informatiche, con l'obiettivo di realizzare una politica comune fra gli Stati sottoscrittori, attraverso l'adozione di una legislazione appropriata, che consenta di combattere il crimine informatico in maniera coordinata. [GL2012]

La Convenzione prevedeva per i Paesi che l'avevano sottoscritta anche un obbligo di adeguamento degli ordinamenti interni, e il nostro Paese, con la legge citata, introduce nell'ordinamento norme di adeguamento proprio al fine di dare piena e intera esecuzione alla Convenzione medesima. [CCCD2016]

Sempre ai fini dell'adeguamento dell'ordinamento interno alle disposizioni contenute nella Convenzione di Budapest, sono state introdotte con la legge n. 48/ 2008 anche numerose novità in materia processuale: sono state modificate alcune norme che disciplinano l'acquisizione degli elementi di prova (ad es. gli articoli 244-247, 248-254, 254 bis, 256, 259, 260, 352, 353, 354 c.p.p.) prevedendo la perquisizione di un sistema informatico o telematico, con misure tecniche dirette ad assicurare la conservazione dei dati originali e a impedirne l'alterazione, e il sequestro di dati, informazioni o programmi informatici utili alle indagini. Sotto il profilo della computer forensics, viene dunque in rilievo il computer inteso non già come strumento del reato ma come strumento utile o rilevante ai fini dell'acquisizione di elementi di prova in relazione a qualsiasi reato direttamente o indirettamente riconducibile all'uso del computer o al sistema informatico, o come "domicilio" di dati, o come "segretezza" di essi, oppure come strumento. [CCCD2016]

L'aspetto più delicato ruota attorno all'esigenza di garantire la genuinità della prova digitale, come già traspariva nella Raccomandazione del 13 settembre 2001 firmata dal Comitato dei Ministri del Consiglio d'Europa, antesignano della Convenzione, nella quale si sottolinea l'esigenza "*to collect, preserve, and present electronic evidence in ways that best ensure and reflect their integrity and irrefutable authenticity (...) should be recognized*"(art. 13) [CM2012]

Il legislatore italiano ha "aggiornato" parte delle disposizioni in tema di mezzi di ricerca della prova allargandone l'oggetto attraverso il riferimento a "sistemi informatici o telematici" prevedendo, in parallelo, che tali operazioni assicurino la conservazioni dei dati originali e la non alterabilità dei dati stessi. Ciò genera un richiamo naturale e necessario alle attività connesse alla disciplina della digital forensics, definita come "*use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations*". [EC2011] Assumono quindi rilevanza e considerazione tutte quelle attività che si sostanziano come serie concatenate di azioni volte alla raccolta di potenziali evidenze elettroniche costituite da bit, con lo scopo di ricostruire eventi penalmente rilevanti, secondo metodi scientificamente approvati e validati.

L'informatica forense ha come fine ultimo quello di conservare, identificare, acquisire, documentare e interpretare i dati contenuti all'interno dei dispositivi, garantendone la non alterazione del sistema informatico già in sede di acquisizione, proseguendo, poi, in sede di analisi attraverso la genuina rappresentazione dei dati così acquisiti: solo in questo modo si potrà "dar voce alle prove" [CM2004]

Le norme tecniche correlate al materiale informatico si fondono a quelle giuridiche nella sfera delle scienze forensi: in altri termini, "il processo di identificare, preservare, analizzare e presentare la prova digitale deve rappresentarsi come accettabile in un procedimento legale o in un contesto legale" [LZ2007] che ai fini di legalità dovrà dirsi conforme alle prescrizioni derivanti dal recepimento italiano della Convenzione di Budapest.

Nel disciplinare il *modus operandi* delle operazioni può osservarsi come l'attenzione del legislatore si sia focalizzata, giustamente, più sul risultato che deve essere ottenuto piuttosto che sul metodo da seguirsi: chiaramente la canonizzazione all'interno di norme

giuridiche di procedure tecniche a livello informatico, più che rappresentare una garanzia, avrebbe sicuramente portato, alla lunga, ad effetti contrari e distorsivi rappresentati dall'evoluzione costante della disciplina e dalle peculiarità proprie di ciascun caso[GC2006]. Fra le righe dell'intervento si scorge un richiamo alle cd best practices del settore, volte a delineare i paradigmi dell'agire tecnico in ambito forense, attraverso una metodologia di base che miri: a) all'**acquisizione** della prova senza alterare o danneggiare il dispositivo originale; b) all'**autenticazione** del reperto e dell'immagine (bit stream image) acquisita; c) a garantire la **ripetibilità** dell'accertamento; d) a un'analisi senza modificazione dei dati originari; e) alla massima imparzialità nell'agire tecnico. [CM2012b]

Prima dell'intervento riformatore del legislatore ad opera della legge n. 48 del 2008, si registravano, purtroppo, prassi non troppo edificanti, spesso mosse da semplificazioni e approssimazioni investigative, che, esaltando fin troppo il ruolo e la portata decisiva assunta dalle digital evidence in alcuni contesti, portavano gli attori mossi dall'entusiasmo ad una crescente disattesa delle procedure, non solo nuove ma anche tradizionali

Nel dettaglio la legge di ratifica consta di 14 articoli, divisi in quattro capi. Tre, in sostanza, sono i campi di intervento finalizzati rispettivamente al rafforzamento degli istituti rilevanti in sede di cooperazione internazionale, ad una miglior armonizzazione e disciplina in ambito di diritto sostanziale relativa al cybercrime e, in ultimo, alla predisposizione di strumenti processuali comuni e condivisi atti all'acquisizione e conservazione delle evidenze elettroniche.

Con la legge n. 48 il legislatore delinea un sistema che appresta non solo un'idonea tutela sostanziale ma altresì dota l'intero sistema processuale di strumenti atti all'acquisizione e valutazione della nascente disciplina sulla prova informatica. Ecco quindi che si fa strada il concetto *digital evidence*, come risultato di attività d'indagine volta sia all'identificazione dell'autore di crimini informatici, **sia all'identificazione dell'autore di reati comuni, commessi col mezzo informatico e non, mediante l'impiego di procedure informatiche proprie della digital investigation**. L'ambito di applicazione del nuovo sistema, infatti, si spinge oltre il terreno del cybercrime, aprendosi a **qualsiasi tipologia di reato per cui si proceda, in perfetta aderenza sul punto con il dettato internazionale della Convenzione, generando quindi un incremento nella domanda d'analisi del dato digitale per fini di ustizia**. [CM2012b]

Così l'art 14 della Convenzione di Budapest: "*... each Party shall apply the powers and procedures referred to in paragraph 1 of this article to: a) the criminal offences established in accordance with articles 2 through 11 of this Convention; b) other criminal offences committed by means of a computer system; c) the collection of evidence in electronic form of a criminal offence*".

L'essenza del provvedimento in materia processuale è rappresentata dalla modifica delle disposizioni relative alle ispezioni e alle perquisizioni: in entrambe i casi viene offerto un "paradigma" sul corretto modus operandi da seguirsi nelle operazioni di accesso al computer oggetto d'indagine, sottolineando la necessità alla salvaguardia dell'integrità dei dati digitali che assume, quindi, canone operativo imprescindibile.

È frequente, infatti, la precisazione circa la necessità di adottare "misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione" a sostegno di un duplice obiettivo: da un lato, garantire la genuina acquisizione di elementi probatori che potranno assumere successivamente valenza di prova, dall'altro, sul fronte delle garanzie difensive, permettere un controllo sull'operato degli inquirenti, il quale deve necessariamente prendere le mosse dalla verifica sulle procedure acquisitive.

La locuzione sopra richiamata lascia "carta bianca"" attraverso un implicito richiamo alle best practices del settore, senza però indicare quale fra le molteplici esistenti ci si debba riferire. Il legislatore mostra una certa indifferenza qualitativa fra le plurime procedure, lasciando margine d'azione e di scelta al "forenser" che concretamente porrà in essere le operazioni secondo la tecnica preferita senza che tale scelta comporti alcuna "trappola della legittimità", sempreché il risultato acquisito rispetti il vincolo della formula normativa sopra richiamata. [CM2012b]

La norma italiana introduce inoltre la richiesta di "preservazione della scena criminis informatica", al 2° comma dell'art. 244 si indica la possibilità per l'autorità giudiziaria di disporre l'ispezione "anche in relazione a sistemi informatici o telematici, adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e la loro inalterabilità"

A tal proposito e nel caso specifico del presente progetto se ne evidenziano le seguenti note:

1. Salvo i casi dove si detengano speciali diritti di accesso, non si ha il potere di alterare i contenuti remoti

2. Il web è dinamico per definizione e non si può assicurare che non avvenga nessuna modifica successiva all'attività; proprio per questa ragione lo strumento software oggetto della presenti tesi si prefigge lo scopo di mantenerne traccia, con un attività di acquisizione forense e con la relativa affidabile conservazione, in contrasto alle potenziali modifiche o eliminazioni del contenuto di interesse originale.

L'attività di acquisizione di contenuti Web rientra a pieno titolo nella categoria degli accertamenti irripetibili, dunque le operazioni ispettive dovranno essere condotte secondo lo schema previsto dall'art. 360 c.p.p. Sebbene l'accertamento possa essere svolto da operatori tecnici in contraddittorio con la parte interessata, eventualmente alla presenza di consulenti tecnici di parte, i risultati così ottenuti saranno cristallizzati in verbali con la conseguente utilizzabilità piena in dibattimento.

2.3 Digital Forensics

Il fine della digital forensics consiste nell'analizzare fatti inerenti sistemi digitali in cui risultino violazioni di leggi civili, penali e/ o di regolamenti interni ad organizzazioni. Se, quindi, non c'è una violazione di qualche tipo, il forensics non interviene. L'analisi condotta deve permettere di ottenere delle fonti di prova digitali, ossia dati che possano testimoniare, nella maniera più certa possibile, i fatti connessi alla violazione da accertare, evidenziando i tempi, gli eventi, i sistemi impiegati, le linee e i mezzi di comunicazione, nonché gli elementi identificativi come nomi utente, IP, MAC address, password, codici biometrici, etc. Le fonti di prova acquisite in tal modo saranno utilizzate quali supporto alle decisioni attraverso il loro ingresso e la discussione in giudizio. Le caratteristiche essenziali della conduzione di una analisi di digital forensics sono essenzialmente tre: 1) impieghi di procedure scientificamente accettate; 2) determinismo e giusta tempistica delle attività svolte; 3) ripetibilità degli accertamenti tecnici. L'impiego di procedure scientificamente accettate (accettabilità scientifica dei metodi) riveste una importanza primaria nel campo dell'analisi; infatti, seguire procedure scientificamente non contestate (e difficilmente contestabili) faciliterà l'impiego dei mezzi di prova in giudizio, mettendo al riparo le evidenze e le risultanze dell'analisi da possibili contestazioni. Allo stato attuale, non esistono delle procedure codificate per condurre una analisi forense, per cui ciascun operatore e ciascuna forza di polizia e/ o ente privato hanno sviluppato dei protocolli interni da seguire, che sono in costante evoluzione e miglioramento.

La digital forensics è una branca delle scienze forensi, ovvero di quelle discipline mediche, biologiche, elettriche, meccaniche, elettroniche, informatiche, etc. che possono fornire evidenze tecno-scientifiche "oggettive" quali elementi di giudizio sia nel caso di procedimenti civili che penali.

Il compito delle scienze forensi è quindi quello di produrre evidenze scientifiche da poter introdurre all'interno di un procedimento giudiziario sotto forma di mezzi di prova, sulle quali il giudice baserà il proprio convincimento e quindi la propria decisione. Si può sicuramente affermare che la digital forensics deve soddisfare sia esigenze di natura tecnico/metodologica che giuridico/legale. La nascita e lo sviluppo della digital forensics come branca indipendente è strettamente correlata all'evoluzione della società dell'informazione (Information and Communication Technology, ICT)

Sul tema non esiste un'omogeneità nella produzione normativa; la disciplina può essere ricondotta ad almeno tre settori di interesse: i crimini commessi attraverso l'uso delle nuove tecnologie (diritto penale dell'informatica), i documenti elettronici e la sicurezza dei sistemi informatici, la tutela dei dati personali. In tutti i casi si tratta di una produzione normativa recente, sviluppatasi nell'arco temporale dell'ultimo ventennio. I reati informatici sono disciplinati nel codice penale e in alcune leggi speciali successive. La convenzione di Budapest del 23 novembre 2001 del Consiglio d'Europa sulla criminalità informatica (ratificata con la legge 48/ 2008) può essere considerata il primo vero accordo internazionale riguardante i crimini commessi attraverso internet o altre reti informatiche, con l'obiettivo di realizzare una politica comune fra gli Stati sottoscrittori, attraverso l'adozione di una legislazione appropriata, che consenta di combattere il crimine informatico in maniera coordinata.

2.3.1 Ripetibilità

Con l'espressione ripetibilità degli accertamenti si intende la possibilità di rieseguire integralmente le analisi su modelli assolutamente identici all'originale; la ripetibilità si configura, quindi, come un fattore chiave soprattutto quando si opera su reati e si devono presentare dei risultati in dibattimento per una incriminazione. In altre parole, si presuppone la possibilità di realizzare uno studio post-mortem dei sistemi, limitando completamente le interazioni con l'esterno. Le fonti di prova digitali validamente ottenute da procedure di digital forensics, dal punto di vista tecnico-scientifico, dovrebbero possedere le seguenti caratteristiche: **integrità, coerenza e documentazione**.

Tutte le analisi di digital forensics si caratterizzano, nella pratica, per il compimento di quattro diverse tipologie di operazioni: l'**identificazione** e il prelievo dei reperti, la **preservazione** e l'archiviazione dei reperti, **l'analisi** dei reperti, la redazione della relazione tecnica contenete i risultati.

Integrità

Per integrità della fonte si intende la possibilità di congelare le evidenze riscontrate in modo da non consentire l'alterabilità nel tempo. Dopo il prelievo della fonte, questa procedura comporta l'adeguata conservazione dei dati.

Coerenza

La coerenza della fonte consiste nel dimostrare che le informazioni assunte durante le analisi di forensics evidenziano fatti logicamente correlati tra loro in una sequenza temporale che sia coerente con le altre informazioni processuali provenienti anche al di fuori dell'ambito digitale.

Documentazione

Per disponibilità della documentazione si intende la procedura documentale certificata che dettaglia tutti i passi inerenti l'estrazione e l'individuazione delle fonti di prova, compresi i metodi e le procedure impiegate, i problemi incontrati, le soluzioni approntate ad hoc, le applicazioni impiegate con relativa versione e licenza, il personale che ha condotto l'analisi con le relative qualificazioni scientifiche, la catena di custodia dei reperti dalla scena del crimine fino al laboratorio, etc.

2.4 Digital Evidence

Si può considerare digital evidence ogni informazione probatoria la cui rilevanza processuale dipende dal contenuto del dato o dalla particolare allocazione su di una determinata periferica, oppure dal fatto di essere stato trasmesso secondo modalità informatiche o telematiche.

Nell'elaborazione dottrinale italiana, per fronteggiare le questioni poste dall'ingresso delle nuove risorse scientifico-tecniche nel processo penale, si è spesso ricorso all'ampio contenitore della prova “atipica”, la cui generica disciplina è fornita dall'art. 189 c.p.p., che consente al giudice l'acquisizione di una prova “non disciplinata dalla legge, se essa

risulta idonea ad assicurare l'accertamento dei fatti e non pregiudica la libertà morale della persona"

2.5 Acquisizione del dato informatico

Nel provvedimento di ratifica ed esecuzione, nel nostro ordinamento, della Convenzione di Budapest sulla criminalità informatica del 2001, il legislatore ha introdotto alcune regole che, ormai da tempo, sono considerate delle best practices. La l. 18 marzo 2008, n. 48, infatti, ruota intorno a due principali perni concettuali:

a) la necessità di misure tecniche che assicurino la conservazione dei dati originali;

b) l'adozione di procedure che non alterino i dati stessi.

Inalterabilità e conservazione dei dati originali saranno il "leitmotiv" del nuovo impianto normativo. La norma di recepimento dell'anzidetta Convenzione internazionale ha espressamente novellato il codice di procedura penale in sette diversi punti:

1. relativamente alle ispezioni, all'art. 244, comma 2, secondo periodo è stato aggiunto: «*anche in relazione a sistemi informatici, telematici, adottando misure tecniche dirette ad assicurare la* ***conservazione dei dati originali e ad impedirne l'alterazione***»;

2. all'art. 247, comma 1-bis, in tema di perquisizioni: «*Quando vi è fondato motivo di ritenere che dati, informazioni, programmi informatici o tracce comunque pertinenti al reato si trovino in un sistema informatico o telematico, ancorché non protetto da misure di sicurezza, ne è disposta la perquisizione,* ***adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione***»;

3. art. 254-bis: «*(Sequestro di dati informatici presso fornitori di servizi informatici, telematici o di telecomunicazioni) l'autorità giudiziaria, quando dispone il sequestro, presso i fornitori di servizi informatici, telematici o di telecomunicazioni, dei dati da questi detenuti, compresi quelli di traffico o di ubicazione, può stabilire, per esigenze legate alla regolare fornitura dei medesimi servizi, che la loro* ***acquisizione avvenga mediante la copia di essi su adeguato supporto, con una procedura che assicuri la conformità dei dati acquisiti a quelli originali e la loro immodificabilità****. In questo caso è, comunque, ordinato al fornitore dei servizi di conservare e proteggere adeguatamente i dati originali*»;

4. art. 256, dedicato al dovere di esibizione e ai segreti, con l'aggiunta della frase «*nonché i dati, le informazioni e i programmi informatici, anche mediante c**opia di essi su adeguato supporto***»;

5. art. 259, in tema di custodia delle cose sequestrate, nel comma 2, è integrato con la frase: «*Quando si tratta di dati, di informazioni o di programmi informatici, **la copia deve esse realizzata su adeguati supporti, mediante procedura che assicuri la conformità della copia all'originale e la sua immodificabilità***»;

6. all'art. 352, in tema di perquisizione, dopo il comma 1 è inserito il seguente: «*1-bis. Nella flagranza del reato, ovvero nei casi di cui al comma 2 quando sussistono i presupposti e le altre condizioni ivi previsti, gli ufficiali di polizia giudiziaria, **adottando misure tecniche dirette ad assicurare la conservazione dei dati originali e ad impedirne l'alterazione**, procedono altresì alla perquisizione di sistemi informatici o telematici, ancorché protetti da misure di sicurezza, quando hanno fondato motivo di ritenere che in questi si trovino occultati dati, informazioni, programmi informatici o tracce comunque pertinenti al reato che possono essere cancellati o dispersi*»;

7. Nell'art. 354, comma 2, in tema di accertamenti urgenti sulle cose e sulle persone, dopo il primo periodo è inserito il seguente: «*In relazione ai dati, alle informazioni e ai programmi informatici o ai sistemi informativi o telematici, gli ufficiali di polizia giudiziaria adottano, altresì, le misure tecniche o impartiscono le **prescrizioni necessarie ad assicurarne la conservazione e ad impedirne l'alterazione** e l'accesso e provvedono, ove possibile, alla loro immediata duplicazione su adeguati supporti, **mediante una procedura che assicuri la conformità della copia all'originale e la sua immodificabilità***».

Dalle sette modifiche poco sopra illustrate si possono ricavare i seguenti principi guida:

a) prestare attenzione a non alterare i dati durante le operazioni di ricerca delle fonti di prova;

b) quando si effettua una duplicazione, assicurarsi che siano garantite la conformità della copia all'originale e la sua immodificabilità.

Corrette modalità di conservazione, procedure di duplicazione efficaci, garanzie di non alterabilità ed extrema ratio del sequestro di servizi, in conclusione, sono i quattro principi della Forensics introdotti nel nostro ordinamento.

2.6 La prova informatica

Il giudice decide l'assunzione della prova in base a quattro criteri: la prova deve essere pertinente, deve cioè tendere a dimostrare l'esistenza di un fatto storico enunciato nell'imputazione; non deve essere vietata dalla legge; non deve essere superflua, cioè non deve tendere ad ottenere un risultato conoscitivo già acquisito e infine deve essere rilevante, ovvero che la sua ammissione determini un reale contributo all'accertamento del fatto da provare. [GL2012] È rilevante inoltre quanto disposto dall'articolo 189 c.p.p., in materia di prova atipica: tale norma prevede che, nel caso in cui venga richiesta l'ammissione di una prova non disciplinata dalla legge, il giudice può assumerla se essa risulta idonea ad assicurare l'accertamento dei fatti e non pregiudica la libertà morale della persona;

2.7 Integrità della prova

Il fatto di lavorare con evidenze digitali ha un enorme svantaggio e un corrispondente vantaggio. I problemi sorgono, come si è già discusso, con l'immaterialità della prova con la quale si ha a che fare. Il dato è aleatorio, perciò è facilissimo corromperlo. Di contro, il vantaggio enorme è che si può copiare il dato quante volte si vuole senza perderne la qualità o senza corromperne la forma. Questo suggerisce di lavorare solo con copie del dato sacrificabili, comportandosi comunque in maniera che il suo intervento sia quanto più trasparente possibile verso la prova.

Si ricorda che in Italia la maggior parte dei casi che coinvolgono un consulente tecnico di Digital Forensics sono assegnazioni di incarico basate sull'articolo 359 c.p.p., che riguardano specificamente accertamenti tecnici ripetibili. È quindi implicito che tutto ciò che l'operatore di digital forensics compie per trovare quanto a lui richiesto debba soddisfare tale condizione sine qua non. Ma proprio perché il si lavora su copie, si potrebbe affermare, senza tema di smentita, che qualunque atto si compia (anche quelli che necessariamente varieranno lo stato della prova) possa essere considerato ripetibile, a meno che non danneggi il supporto originale o la prima copia (quella acquisita in fase di sequestro). [FG2013]

Da qui la necessità di corredare ogni acquisizione sempre con la firma hash (md5, ma preferibilmente sha1) di ogni singolo elemento acquisito, così da poterne fornire le adeguate garanzie di integrità.

2.8 Conclusioni

Si sono espresse in questo capitolo le principali "categorie" che meglio possono rappresentare le attività "da contrastare sul Web" e che lo strumento software oggetto della presente tesi avrà l'onere di acquisire con l'obiettivo di congelarne la prova informatica da sottoporre al successivo utilizzato in sede di giudizio.

Ovviamente quanto appena esposto non è, e non sarebbe potuto essere, un panorama completo di tutto ciò che realmente può essere definito "reato di interesse informatico". Inoltre, anche volendo perseguire l'obiettivo di arrivare ad una catalogazione completa, risulterebbe comunque uno sforzo inutile in quanto tutti i reati ad oggi previsti dal codice penale, o le condotte illecite di natura civile, se compiuti lasciando un traccia digitale, risulteranno di interesse nelle attività forensi di acquisizione di prove informatiche, siano essi **reati informatici "in senso stretto"** , ovvero quelle particolari figure di reato in cui il profilo informatico (e quindi la connessione, l'elaboratore, il software ecc.) si presenta come imprescindibile elemento della condotta o dell'evento del reato o **reati informatici "in senso lato"**, ovvero tutte quelle fattispecie di reato "comune" che, per le particolari modalità con cui viene posto in essere, si presta ad implicazioni di carattere informatico, ma in maniera del tutto accidentale e assolutamente non caratterizzante.

Dopo questa prima disamina dal sapore giuridico, si passerà nel prossimo capitolo a dare invece una panoramica più tecnica che possa meglio delinearne il contesto scientifico e tecnologico nel quale il prodotto software oggetto della presente tesi dovrà inserirsi.

3 Aspetti tecnologici dell'acquisizione di contenuti nel web moderno

Negli ultimi venti anni l'utilizzo di Internet ha avuto un notevole impatto nella società moderna. Lo spazio cibernetico ha aperto certamente nuove strade per lo sviluppo economico ma la digitalizzazione delle comunicazioni e lo scambio di informazioni porta con se il pericolo di furto, contraffazione e alterazione dei dati nello spazio cibernetico che può, al tempo stesso, essere altresì il luogo ideale per l'occultamento delle tracce e dei proventi dei reati. Alla luce della proliferazione della criminalità informatica e nel contesto di una continua evoluzione tecnologica a cui si accompagna una delocalizzazione dei dati e delle informazioni, appare indispensabile riflettere sui mezzi e gli strumenti che possano permettere agli operatori un efficace contrasto alla criminalità (di ogni livello, dalla criminalità internazionale al crimine compiuto da una persona qualunque) che si avvale dello strumento informatico e della "rete" in particolare; in questa ottica certamente assume un ruolo centrale la tematica dell'individuazione della prova elettronica (e-evidence) e dei mezzi di ricerca e acquisizione della stessa in un contesto, quello digitale, ontologicamente rarefatto a cui con difficoltà possono adeguarsi i tradizionali paradigmi dell'attività investigativa. [CCCD2016].

3.1 Web moderno e barriere di accesso al dato

Col passare del tempo e della relativa evoluzione del Web e delle sue tecnologie, il panorama si è progressivamente ampliato introducendo nuove complicazioni. Nel Web di qualche anno fa i contenuti erano prevalentemente "statici", le pagine che

prevedevano comunicazioni asincrone erano l'eccezione e le poche esistenti sfruttavano tecniche grezze e poco più che sperimentali.

Il "vecchio" Web era un mondo più "semplice", non solo tecnicamente, ma anche per le interazioni umane. Il panorama si presentava quindi decisamente più "ingenuo" rispetto a quello attuale, e non vi era una costante necessità di introdurre sofisticate barriere di accesso al dato. Fare il "clone" (nel nostro caso acquisizione) del contenuto di interesse risultava un'attività che presentava meno complicazioni rispetto ad oggi. Erano tempi dove il solo comando *wget* poteva risultare sufficiente, riuscendo a svolgere tutto in semplicità producendo in output un lavoro completo e privo di lacune. Oggi, invece, con l'evoluzione sempre più dinamica e asincrona dei contenuti, con inoltre l'assidua presenza di minacce, è sempre più intricato arrivare ad acquisire la prova digitale presente sul Web con soli pochi e semplici comandi.

Le difficoltà che il "nuovo" Web pone davanti, non sono introdotte solo dall'evoluzione delle nuove tecnologie, ma anche dalla presenza di sempre più attente precauzioni tecnico-legali poste innanzi al dato da chi ne detiene la proprietà. Queste barriere sono vere e proprie contromisure espressamente poste in contrasto alle minacce che possono affliggere dati e utenti, che sempre più spesso corrono il rischio di vedersi oggetti di malintenzionate attenzioni con l'obiettivo di estirpare informazioni per usi non sempre leciti (phishing, data leak, profilazione, etc).

Inoltre tra le barriere si possono anche annoverare tutte quelle informazioni fruibili solo in presenza delle necessarie caratteristiche capaci di soddisfare determinate policy di privacy potenzialmente presenti su di un determinato contenuto. Come per esempio tutte quelle risorse che un utente pubblica su di un Social Network e che risulteranno accessibili solamente alla sua sfera di contatti. In questo esempio, per accedere al dato risulterà necessario avere un collegamento di primo grado con quello specifico utente.

Vi sono poi quei casi dove per accedere al contenuto di interesse viene richiesto all'utente di provare la propria appartenenza al genere umano, così da contrastare l'avanzata dei "*bot*" che sempre più assiduamente minacciano i dati. Strumenti utili a contrastarli sono gli ormai ben noti "captcha" che, in una crescente opera di creatività, si sono evoluti da "banali" controlli basati su "semplici" testi deformati a vere e proprie analisi comportamentali sulle attività compiute all'interno del browser.

Tutti i predetti blocchi di accesso al dato (tra cui possiamo includere i banner modali di accettazione dei cookie, richieste di autenticazione, token inviati su header HTTP

arbitrari, etc), risultano decisamente di difficile, e in alcuni casi anche di impossibile superamento con gli strumenti software "vecchio stampo" che mai hanno previsto nei loro scopi la necessità di contrastare le predette barriere introdotte dal Web moderno, risultando quindi ormai inutili per gli scopi di nostro interesse.

Risulta quindi evidente come tutto ciò apra una pletora di casistiche non sempre di facile gestione, per le quali si evidenzia la necessità di nuovi e sempre più flessibili strumenti, in grado di essere al passo con lo stato dell'arte del Web e della sua continua evoluzione.

Di seguito si proporranno due macro categorie principali per le tipologie di contenuti di nostro interesse:

- Soluzioni per l'acquisizione forense di contenuti statici
- Soluzioni per l'acquisizione forense di contenuti complessi/dinamici

3.2 Soluzioni per l'acquisizione forense di contenuti statici

Come precedentemente menzionato, molte delle soluzioni dedicate all'acquisizione di un contenuto Web (seppur non strettamente forensi) sono studiate per affrontare un WWW ormai superato, presentando un approccio prevalentemente statico e quindi prive di quelle caratteristiche che gli permetterebbero di acquisire quegli stream articolati che ad oggi compongono la grande maggioranza dei contenuti presenti in Rete.

Oltre ai classici strumenti standalone da eseguire in un computer locale (es: wget, curl), ne esistono molti altri accessibili online, capaci di "congelare" il contenuto di interesse in una precisa occasione temporale. Il risultato fornito da questi strumenti è reso solitamente fruibile tramite la generazione di un indirizzo univoco che sarà quindi consultabile successivamente da un normale browser, permettendo di verificarne sia il contenuto, sia i riferimenti temporali di quando è stata svolta l'attività di "congelamento" richiesta.

Seppure in un primo momento questi strumenti potrebbero assomigliare, per lo meno nei modi, a quanto si prefigge il presente progetto, cioè ad un servizio di acquisizione fruibile via web che preveda la conservazione del dato e la relativa fruizione di quanto acquisto, in realtà mostrano lacune che ne vanificano l'utilizzo come strumento forense.

In particolare:

- mancanza di chiarezza sulle metodologie utilizzate per l'acquisizione del dato

- assenza di ogni informazione in merito alle garanzie di custodia del dato e della sua integrità
- mancanza delle firme hash di quanto acquisito
- impossibilità di esportare i contenuti acquisiti per rendere possibile una consultazione "offline" che, anche quando presente, risulta comunque incompleta di documentazione, firme hash e log delle richieste HTTP.

Quindi, seppur ottimi strumenti che hanno come obiettivo quello di tenere traccia del passato del Web, risultano per lo più inutili in ottica forense, non fornendo tutte quelle caratteristiche che un'acquisizione di tal genere richiederebbe, quali la descrizione delle metodologie utilizzate e i relativi dettagli tecnici, la memorizzazione distinta delle risorse acquisite, la firma hash dei contenuti, i riferimenti temporali certi ed eventualmente fornire la possibilità di una "*catena di custodia*" (affidabile).

A fronte di queste lacune non marginali e della crescente rilevanza probatoria del Web, i singoli operatori forensi (sia forze dell'ordine che consulenti tecnici) si sono visti costretti ad adottare di volta in volta soluzioni tecniche arbitrarie, con il rischio di avere in mano acquisizioni di dubbia attendibilità tali da compromettere l'attività svolta e, sopratutto, mettere a rischio il processo giuridico interessato. Non è quindi nuovo il problema di adottare strumenti e tecniche che possano portare il giusto contributo alla causa forense senza dover ogni volta "reinventare la ruota". Per assurdo, solo ultimamente il panorama tecnico ha portato sul campo strumenti software atti a fornire una soluzione "forensics oriented" a questo tema così importante e delicato.

Risulterà quindi necessario fornire uno strumento capace di acquisire il più ampio numero possibile di risorse veicolate tramite Internet, sia per i classici contenuti "vecchio stile" (statici) prettamente HTML, con poco (se non del tutto assente) contenuto asincrono e relativamente scarso utilizzo di script client-side, sia per quelli moderni che invece sempre più spesso utilizzano script client-side anche allo scopo di impaginare il layout (Es: AngularJS). Risulterà però fondamentale implementare un sistema di acquisizione in grado di "congelare" proprio quei contenuti (dinamici) veicolati tramite le tecnologie più attuali, proprie del così (tristemente) detto "Web 2.0", che proprio a causa di un massiccio utilizzo di contenuto asincrono, oltre che elevato utilizzo di componenti basati su script client-side, risultano particolarmente sfuggenti e quindi di difficile acquisizione con strumenti classici di "mirroring", più adatti per i contenuti di tipo statico.

Di seguito vengono riportate varie soluzioni on-line che permettono attività di "congelamento" di contenuti Web prevalentemente statici:

- **Hashbot** (https://www.hashbot.com/)

  prodotto italiano, unico dei qui presenti strumenti "web-based" elencati nella categoria "statica", che ha il dichiarato obiettivo di generare un risultato in ottica forense, generando gli hash dei contenuti e rilasciando i log HTTP di quanto acquisito, fornendo poi l'archivio zip del materiale elaborato che sarà poi eliminato dal server dopo 120 secondi dal momento dell'acquisizione.

  I risultati ottenuti prevedono il solo salvataggio della risorsa richiesta, quindi in caso di una pagina HTML il solo codice sorgente senza i contenuti aggiuntivi presenti in essa.

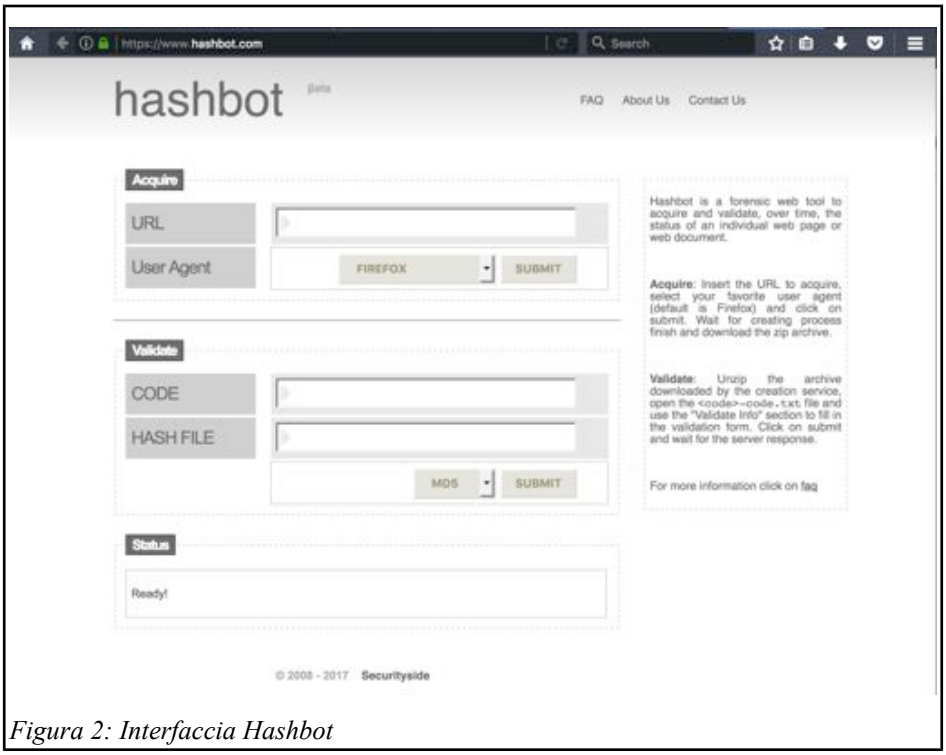


Figura 2: Interfaccia Hashbot

- **Internet Archive - WayBack Machine** (https://web.archive.org)

  storico servizio web based che tiene la storia del Web effettuando snapshot automatici delle pagine e organizza gli stessi in una linea temporale. Oltre al suo normale workflow di indicizzazione storica, è possibile "forzare" uno snapshot della pagina che sarà così archiviato nella memoria collettiva del servizio.

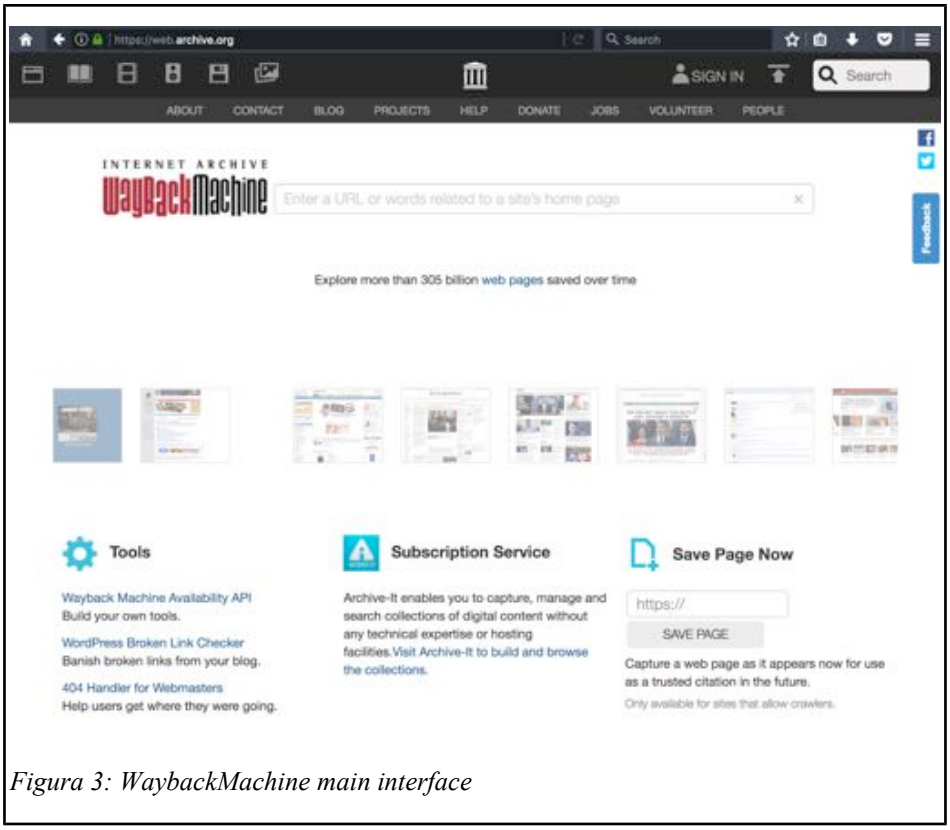


Figura 3: WaybackMachine main interface

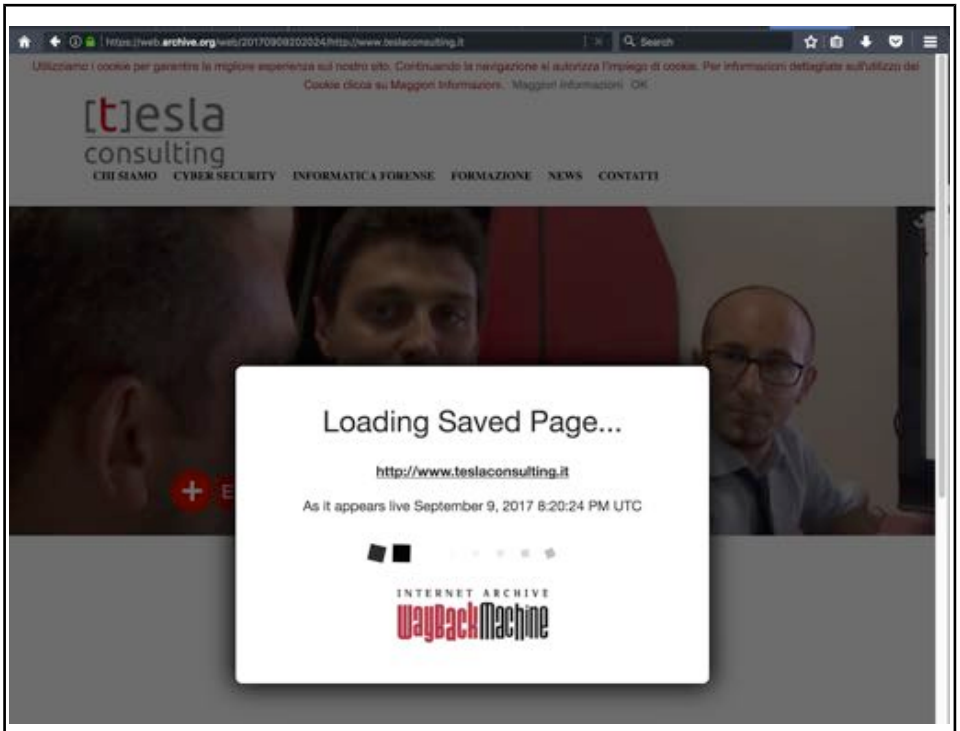


Figura 4: WaybackMachine salvataggio

- **archive.is - webpage capture** (https://archive.is):

  servizio web based che permette di creare snapshot degli url di interesse, i quali saranno archiviati pubblicamente ognuno accompagnato con i relativi riferimenti temporali

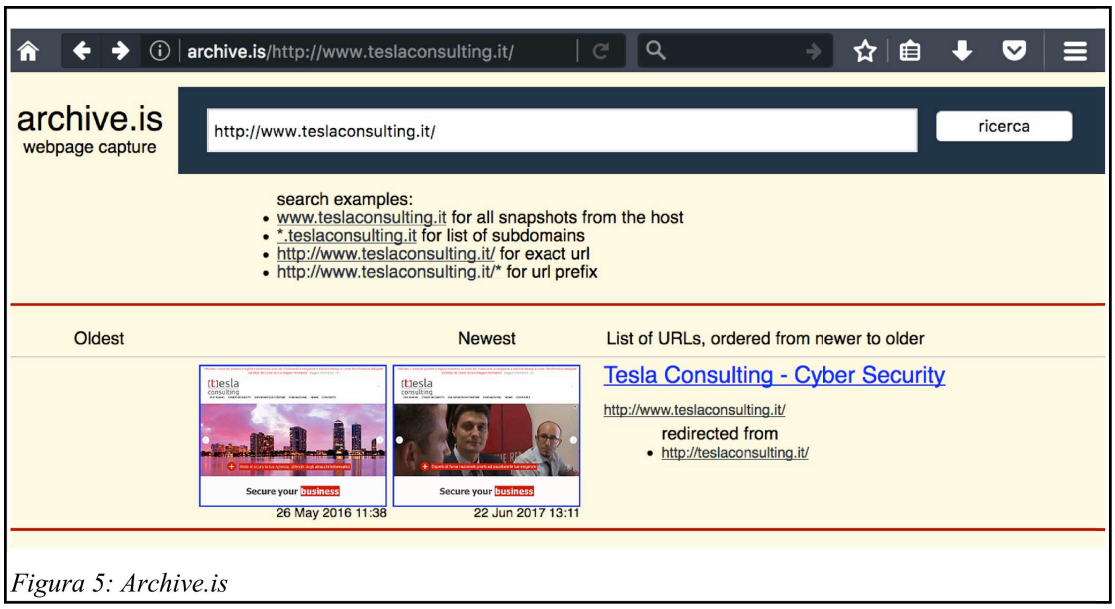


Figura 5: Archive.is

- **WebCite** (https://www.webcitation.org):

  servizio di archiviazione "on-demand" simile ai precedenti che permette di "congelare" le pagine in un dato momento così da poterne tenere memoria storica. A differenza degli altri servizi analoghi permette di aggiungere dei metadati aggiuntivi relativi allo snapshot che si sta effettuando.

Figura 6: WebCite

- **wget/curl** (o altri tool analoghi):

  non sono soluzioni web-based, ma software "classici" installati su un sistema operativo ospitante. Offrono la possibilità di effettuare il download delle risorse sia singole che ricorsivamente in base al contenuto HTML, potendo quindi effettuare il "mirroring" dei contenuti statici di un intero sito.

  Di per sé non sono strumenti forensi, ma lo possono diventare se utilizzati in maniera consona da professionisti consapevoli delle necessità, come per esempio effettuare il calcolo delle firme hash di ogni risorsa scaricata e tenere traccia di tutte le operazioni svolte. Risulta inoltre necessario tutti i log dell'applicativo, infatti questi strumenti offrono molti dettagli (non disponibili in tutti i tool precedentemente menzionati), tipo gli status e gli header del protocollo HTTP, potendo quindi indagare su vari dettagli, come per esempio la data di ultima modifica del contenuto, o se esso è in realtà rediretto su altro server.

3.3 Soluzioni per l'acquisizione forense di contenuti complessi/dinamici

Così come appena accennato, si sono quindi da poco affacciati nel panorama forense alcuni strumenti che cercano di aiutare a superare le difficoltà poc'anzi accennate e tra questi FAW e LegalEye si possono sicuramente annoverare tra i più completi nel panorama forense.

- **FAW - Forensics Acquisition of Websites** (http://it.fawproject.com/):

  browser *freeware* dedicato alle acquisizioni forensi da installare localmente su sistemi Windows (viene offerta anche una macchina virtuale pronta all'uso)

- **LegalEye** (https://www.legaleye.cloud/):

  servizio a pagamento in *cloud* di acquisizione forense di contenuti web basato su strumenti proprietari che permettono la creazione di una sessione desktop eseguita in una macchina virtuale remotizzata e resa fruibile tramite un normale browser.

Le due precedenti soluzioni sono state anche punto di riferimento di partenza del presente progetto e saranno trattate maggiormente in dettaglio successivamente, quando

verranno effettuati i confronti tra le due soluzioni con il progetto oggetto del presente documento

I due progetti appena menzionati saranno di seguito trattati cercando di darne una panoramica ad alto livello che possa successivamente rendere chiari i punti di contatto con il progetto in esame della presente tesi, il quale tenta di amalgamare le proprie idee originali con le influenze indotte dagli aspetti più interessanti dei due applicativivi sopra citati.

3.3.1 FAW – Forensics Acquisition of Websites

[http://www.fawproject.com/]

Arrivato alla versione 6.2.0.0 al momento della stesura del presente documento, viene definito dai propri creatori come il “primo browser forense per l'acquisizione di pagine web”.

Come dice la sopra menzionata definizione FAW è un browser web gratuito (closed source) sviluppato in .NET integrante il motore di rendering di Chromium (Blink). Primario obiettivo di FAW è effettuare acquisizioni forensi potendo sfruttare un workflow ormai ben consolidato come quello di una normale navigazione web; FAW, come si può intuire dalla definizione stessa risulta essere un applicativo client side da installare su un sistema operativo ospitante di tipologia Windows, quindi non pensato per essere utilizzato remotamente e richiede quindi la predisposizione di un ambiente proprietario e licenziato sul quale poter essere eseguito.

Questo ultimo aspetto non è di certo secondario, in quanto seppure il prodotto risulta essere decisamente un progetto interessante e ben strutturato, lascia l'onere a proprio carico di creare un ambiente il più possibile incontaminato, affidabile e a prova di qualsiasi ragionevole dubbio che possa evitare du mettere a rischio la prova. Questo è quindi sicuramente un punto critico che non facilita il lavoro degli operatori forensi che dovranno farsi carico non solo della fase di acquisizione, ma anche della creazione dell'ambiente operativo ripresentando fortemente il problema, di cui si menzionava anche nel capitolo precedente, di creare soluzioni arbitrarie lasciate all'inventiva del singolo operatore, generando di volta in volta soluzioni anche molto differenti tra loro da valutare caso per caso con il solito rischio di perdere la prova per ingenuità non ben valutate in fase di predisposizione dell'ambiente.

A tal proposito il progetto FAW cerca di arginare la problematica rilasciando una macchina virtuale basata su Windows con già tutto l'occorrente installato e pronta all'uso. Questo però pur facilitando la predisposizione dell'ambiente non risolve tutte le possibili problematiche legate all'ambiente fisico ospitante, configurazioni di rete che potrebbero far insorgere ragionevoli dubbi sulla modifica del dato nel segmento di rete che va dalla macchina virtuale al router perimetrale e non per ultimo il licensing "border line" del sistema operativo preinstallato sulla macchina virtuale.

Caratteristiche:

Il prodotto si caratterizza per essere pensato come uno strumento di acquisizione per riuscire ad effettuare acquisizioni di contenuti complessi quali pagine con frame (anche annidati), pagine contenenti stream audio/video, pagine con contenuti asincroni. Prevede inoltre l'estrapolazione completa in fase di acquisizione dei vari contenuti grafici connessi alla pagina oltre che il codice HTML e i relativi script e css e gli header HTTP di ogni componente scaricata.

Altri aspetti interessante in ottica forense sono:

- integrazione con WireShark che permette anche l'acquisizione dello stream del flusso di rete a più basso livello
- cattura video della sessione utente, così da rendere esplicita e comprensibile l'attività anche ai non tecnici che difficilmente sarebbero in grado di apprezzare a pieno certi tecnicismi.
- calcolo automatico degli hash sha1 e md5 di tutti i file acquisiti
- riepilogo dell'acquisizione in un file di riepilogo con un log dettagliato di tutte le operazioni effettuate, files creati ed orari, "identificando" l'autore dell'analisi tramite IP e identificativi univoci della macchina.

Limiti:

Il primo evidente limite di FAW è quella di essere un software locale con la necessità di essere installato in autonomia su un sistema ospitante e con su propria infrastruttura di rete, che ne evidenzia, come già precedentemente menzionato, tutti i suoi limiti come soluzione forense in quanto in caso di non corretta configurazione anche di una sola componente ci sarebbe il serio rischio ci mettere in seria discussione la prova con il rischio di comprometterla irrimediabilmente.

Altro limite è l'essere mono sistema operativo, obbligando l'operatore ad avere necessità di un prestabilito sistema limitandone di conseguenza le possibilità di scelta infrastrutturale.

Lacune riscontrate in FAW durante la fase di acquisizione:

- assenza del codice “raw” ottenuto dalla richiesta HTTP principale, fornendo come output dell'attività di acquisizione solamente il codice ottenuto dalla post-elaborazione del DOM da parte degli agenti attivi (script) della pagina HTML.
- mancanza del tracciamento dei redirect HTTP necessari per raggiungere le risorse acquisite
- acquisizione dei soli header HTTP della pagina principale e relativa assenza degli header HTTP delle risorse scaricate e in caso di presenza di frame assenza dei log HTTP delle risorse puntate dai frame (dei quali viene fornito solo il codice HTML post elaborazione di rendering).
- incapacità di riprodurre formati video h264
- se in presenza di traffico HTTPS questo non sarà consultabile nel dump di rete effettuato.

3.3.2 LegalEye

[https://www.legaleye.cloud/]

Si tratta di un sistema server-side proprietario, basato su piattaforma Citrix in congiunzione ad un sistema Microsoft Windows, che permette agli utenti di collegarsi su di un'istanza di una macchina virtuale remotizzata che viene rigenerata da zero ad ogni nuovo caso di acquisizione.

Essendo un sistema a pagamento, l'accesso alla macchina virtuale viene consentita solo dopo aver effettuato login con il proprio account usufruendo di un normale Web browser. Dopo la fase di login, sempre tramite una normale sessione di navigazione, si potrà accedere all'istanza del sistema remotizzato dove tutta l'attività dell'operatore sarà tracciata tramite acquisizione del traffico di rete e registrazione video del desktop.

Non esiste una vera e propria fase di acquisizione forense, come nel caso del precedentemente menzionato FAW, in quanto l'operatore avrà a disposizione una normalissima installazione di Mozilla Firefox sul quale è predisposto un add-on che permette di effettuare in autonomia gli screenshot della pagina tramite un “click” sul corrispondente pulsante, mentre la fase di acquisizione del codice HTML è lasciata all'operatore che dovrà semplicemente effettuare una normale operazione di file: salva con nome.

A fine attività viene rilasciato il traffico di rete generato, la sessione video registrata, gli screenshot comandati, i salvataggi effettuati e lo snapshot stesso della macchina virtuale.

Limiti:

Da un punto di vista forense il più grande limite è la mancanza di una fase di acquisizione messa su binari stringenti in grado di acquisire in maniera molto ferrea quanto si vuole acquisire. Lasciando all'operatore di effettuare un normale salvataggio “salva con nome” dal browser, l'output che si otterrà non sarà pedissequamente uguale all'originale, in quanto il browser cerca di generare un contenuto fruibile offline, modificando all'occorrenza il codice HTML per renderlo di più immediata consultazione visiva.

Non viene quindi generato un vero e proprio codice HTML “puro” del DOM, così come per FAW non viene acquisito il codice HTML “raw” al momento della prima richiesta HTTP.

Inoltre non sono riassunti gli header HTTP ne della pagina acquisita ne delle singole risorse, ma viene tutto lasciato come onere di consultazione tramite il traffico acquisito, il quale, come per FAW, in caso di traffico SSL/TLS non sarà interpretabile in quanto cifrato.

3.4 Conclusioni

Ad oggi le due soluzioni menzionate sono il fiore all'occhiello dello stato dell'arte dell'acquisizione forense di contenuti Web e come appena menzionato non mancano di certo limiti e imperfezioni.

Forse i limiti appena menzionati delle sopra citate soluzioni possono suonare un po' troppo severi, ma alla luce dei fatti sono molto concreti se visionati nel dettaglio.

- **mancanza del codice “raw”**: gli strumenti precedentemente menzionati, non hanno questa caratteristica che risulta però fondamentale in tutti quei casi come per esempio attività di “incident response” post infezione di malware bancario, dove è necessario riscontrare come una data infezione può essersi propagata e tale indizio può del tutto essere assente nel codice HTML post elaborazione del DOM, dove l'agente malevolo stesso può rimuovere le sue stesse tracce
- **incapacità di analizzare traffico HTTPS**: è un limite importante, in quanto si rischia di rendere nullo buona parte del traffico acquisito non potendo aver modo di ispezionarlo e può essere sia un problema formale che di indagine in quanto può risultare attività sensibile per le analisi fare un'ispezione di quanto si è veicolato. Risulta un problema formale in quanto si viene a perdere tutta la correlazione tra quanto acquisito e il traffico stesso, quindi verrebbe a mancare il formalismo stesso per avere in mano dei dati certi, risulta invece essere poi un problema per le indagini quando l'attività è volta a contrastare il “cyber-crime” che usa veicolare dati tramite HTTPS ormai come pratica consolidata e poter analizzare il payload malevolo da di un contenuto veicolato da un sito risulta essere un'attività importante per determinare il reato informatico intercorso, così come indicato nel capitolo inerente alle frodi informatica e phishing.

- **mancanza di tracciamento dei redirect HTTP**: Tracciare in maniera puntuale i redirect risulta una cosa molto importante in tutti quei casi dove la risorsa puntata (per esempio un video) è riferita inizialmente ad un dato indirizzo e poi ridiretta su altri servizi sul cloud. Questo può risultare particolarmente determinante per poter identificare la reale ubicazione della risorsa incriminata, le eventuali responsabilità e la relativa supremazia statale di competenza oltre che riuscire a determinare in maniera certa a chi inviare istanza di perquisizione e/o confisca nei luoghi di ubicazione reale del dato.

Meno rilevanti da un punto di vista forense, ma che introducono comunque dei punti delicati, sono i limiti di piattaforma poc'anzi accennati:

- Nel primo caso (FAW) si crea un vincolo ad una piattaforma specifica, il che può introdurre problemi operativi nel caso non si sia nelle disponibilità di poterne usufruire o per costi o per semplice incompatibilità con l'infrastruttura di supporto a disposizione.
- Nel secondo caso (LegalEye) i limiti di piattaforma hanno delle ripercussioni sui costi di implementazione in quanto si avrà la necessità di dotarsi di licenze particolari (e non sempre economiche) e di conseguenza ci sarebbe una ripercussione sui costi di ogni singola acquisizione e tale costo ricadrebbe inevitabilmente anche sugli utenti finali.

Scopo dell'attuale progetto di tesi sarà quindi quella di creare uno strumento forense di acquisizione di contenuti Web e che possa aggiungersi al panorama presente potendo offrire caratteristiche ad oggi ancora non o presenti in parte in altri prodotti.

L'approccio tecnico sarà quello di basare la soluzione su tecnologie Open-Source con lo scopo ulteriore di rendere il prodotto fruibile su qualsiasi piattaforma proponendo la fruizione della soluzione tramite piattaforma Web-based.

La soluzione che si andrà a descrivere è ad oggi sicuramente in uno stato acerbo e non completa di tutte le caratteristiche che dovrà possedere il prodotto finale, complice anche il fatto che durante la fase di sviluppo si sono attuati cambi di rotta radicali e introdotte nuove idee, rendendo il progetto assai differente e anche più ampio rispetto agli intenti iniziali.

4 Webidence - un sistema di acquisizione forense di contenuti Web dinamici

Come spesso accade, si arriva ad un punto in cui può risultare necessario gettare tutto il lavoro fatto (o quasi) a causa dei limiti intrinsechi delle soluzioni tecniche scelte. Ma non è detto che arrivare ad un punto morto risulti essere una cosa negativa, è anzi importante rendere esso stesso una parte integrante dell'attività di studio, in grado di evidenziare debolezze ed eventuali impossibilità materiali a proseguire per tale strada. Così lo studio e lo sviluppo del presente progetto ha avuto due fasi di vita ben distinte, ognuna caratterizzata da scelte tecniche e approcci molto differenti tra loro. Di seguito si tratteranno entrambe le soluzioni adottate, partendo dall'illustrazione della prima versione (poi scartata) indicandone i limiti riscontrati e le relative motivazioni che hanno successivamente portato alla decisione di virare drasticamente sulla seconda soluzione, divenuta poi la scelta finale sulla quale proseguire.

4.1 Fase 1 (Webidence 1.0)

4.1.1 Panoramica

La prima fase aveva come obiettivo lo sviluppo di uno strumento di acquisizione forense di contenuti Web che potesse essere veicolato completamente da sole richieste HTTP, mediate da una web application server side intermedia che svolgesse per un certo verso il ruolo di "proxy". Così facendo l'utente, senza nulla configurare sul proprio browser, si sarebbe trovato davanti, in maniera del tutto trasparente, il contenuto di interesse da acquisire, senza che al suo occhio risultasse la presenza di un "proxy

intermedio". Unica traccia della presenza di un "proxy" risultava essere l'introduzione di una "barra di navigazione" in cima alla pagina, così da poter creare la metafora del "meta-browser annidato". Questa metafora (che è poi rimasta invariata nella seconda fase del progetto) aveva lo scopo di ricreare un ambiente famigliare che permettesse all'operatore di ritrovarsi innanzi ad uno strumento noto e consolidato, senza quindi la necessità di sovraccaricarlo di nuove nozioni.

Per rendere l'interfaccia così "auto-magica" e semplificare quindi l'attività dell'utente, si è dovuto "nascondere" dietro al cofano dell'applicazione, lato server, tutte quelle complicazioni che progressivamente si esponevano, per così rendere la richiesta dell'utente fruibile senza imperfezioni che ne ostacolassero la consultazione. In particolare, gli ostacoli più grandi si sono riscontrati verso i servizi più complessi e che, più o meno velatamente, inserivano controlli mirati a interferire e contrastare proprio la fruizione dei contenuti al di fuori dei canali ufficiali. Pagine come facebook, twitter, google, (ma anche altre) sono risultate essere particolarmente problematiche, tanto da richiedere soluzioni "hard-coded" personalizzate per ogni singolo ostacolo esposto da ciascuna di esse. E, seppure in grado di rendere le singole pagine fruibili, le personalizzazioni introdotte penalizzavano il progetto rendendolo particolarmente sensibile alle perturbazioni imprevedibili di ciascuna pagina.

Pur con i limiti presenti in questa prima realizzazione, non sono invece stati riscontrati particolari problemi su tutte quelle pagine quali forum, siti fatti in Worpress, Joomla, pagine vetrina, etc., cioè pagine dalle dinamiche decisamente più agevoli e lineari.

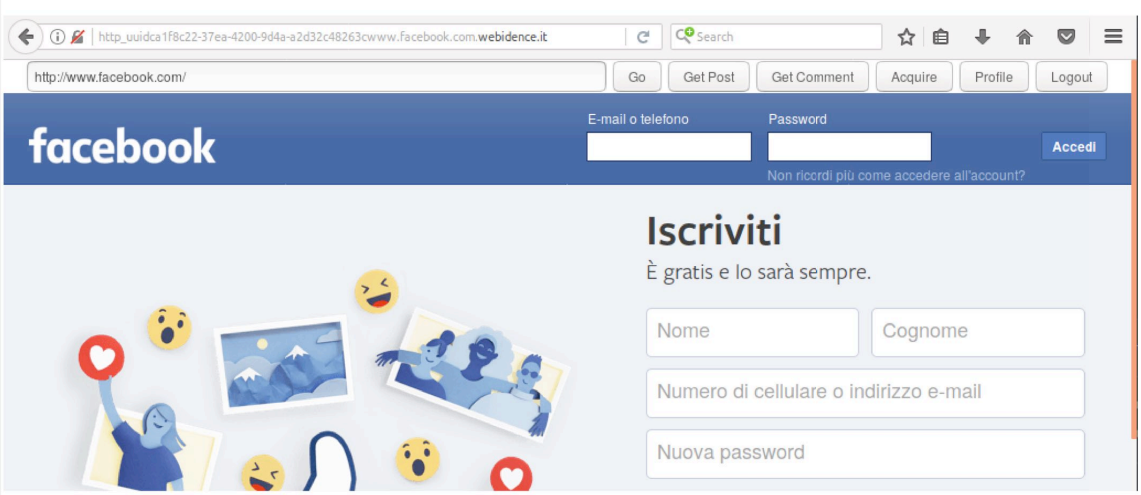


Figura 7: Interfaccia annidata

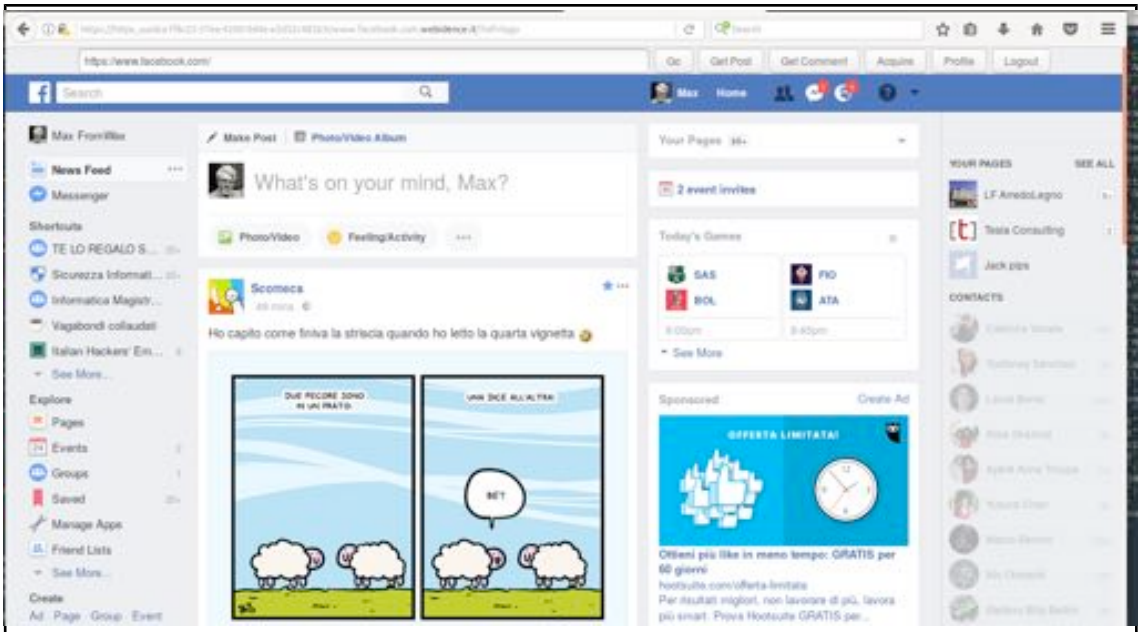


Figura 8: Esempio di supporto al login annidato

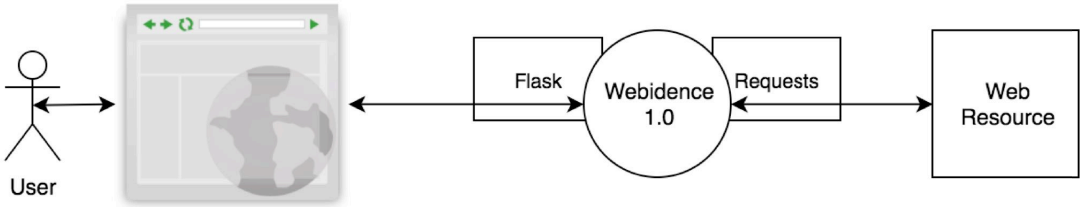


Figura 9: Webidence 1.0 "user interaction" di base

Come precedentemente anticipato, l'interfaccia utente proposta è stata progettata con l'intento di riprodurre un meta-browser annidato che permettesse all'utente di svolgere una “normale” navigazione verso la risorsa di interesse, nella maniera più agevole e famigliare possibile.

Proprio grazie alla predetta caratteristica, si rendeva possibile la fruizione di contenuti accessibili solo mediante autenticazione, senza la delicata necessità di memorizzare le credenziali utente. E ciò era reso possibile attraverso l'intercettazione dei cookie di sessione (e in generale di qualsiasi header HTTP) e il loro successivo salvataggio, così da poterli riciclare nella fase dedicata alla vera e propria acquisizione del contenuto di interesse. Si rendeva il tal modo possibile garantire all'utente la dovuta attenzione alla sicurezza delle sue informazioni.

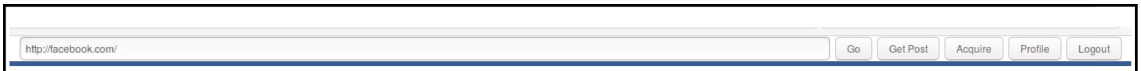


Figura 10: Webidence – la URL bar su facebook.com

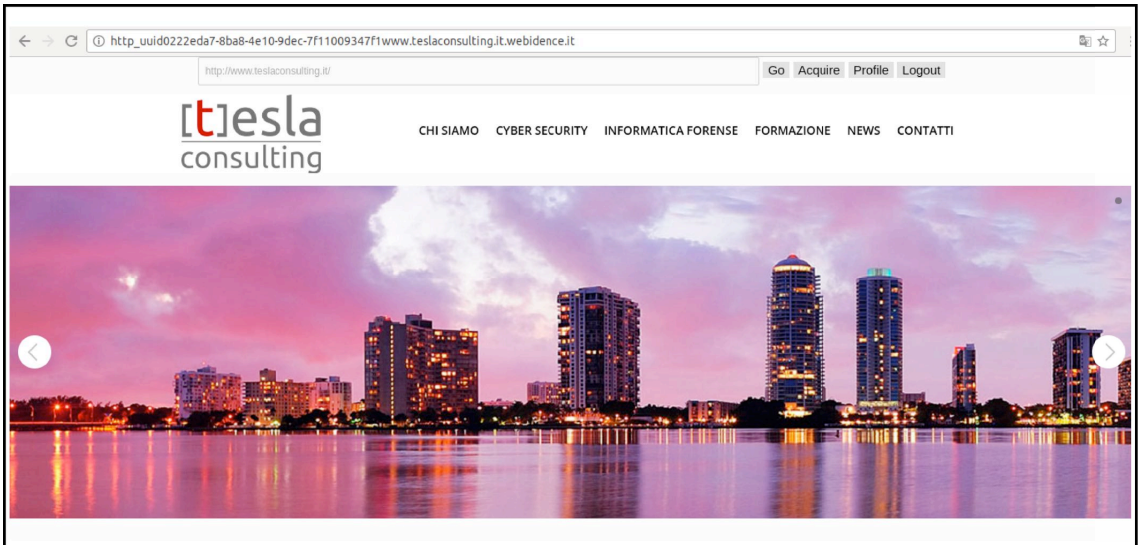


Figura 11: Webidence - URL bar di base per siti "normali"

La barra di navigazione del meta-browser contiene al suo interno vari pulsanti che permettono di compiere specifiche operazioni e alcuni sono contestuali al dominio in cui ci si trova:

- **pulsante "GO"**: avvia la meta-navigazione all'indirizzo inserito sulla meta-addresBar dall'utente
- **pulsante "Profile"**: porta alla pagina profilo dell'utente dove sarà possibile visionare lo stato delle proprie acquisizioni, il credito residuo, i dati personali
- **pulsante "Get Post"**: questo pulsante sarà presente contestualmente alla meta-navigazione su facebook.com e la sua (ovvia) funzionalità è quella di poter acquisite un singolo *post.* Una volta selezionato il pulsante si attiverà la modalità legata a tale funzionalità e il cursore prenderà la forma di un mirino (+), successivamente al passaggio del mouse su uno dei post presenti sulla timeline di facebook, questi sarano evidenziati da un bordo rosso e al click su di esso sarà messa in coda l'acquisizione del post scelto (Figura 12 e Figura 13)
- **pulsante "Acquire"**: come facilmente intuibile il pulsante "acquire" è il nucleo principale del progetto e, una volta cliccato, eseguirà l'inserimento nella coda delle acquisizioni dell'url che in quel momento l'utente starà visionando con il meta-browser.

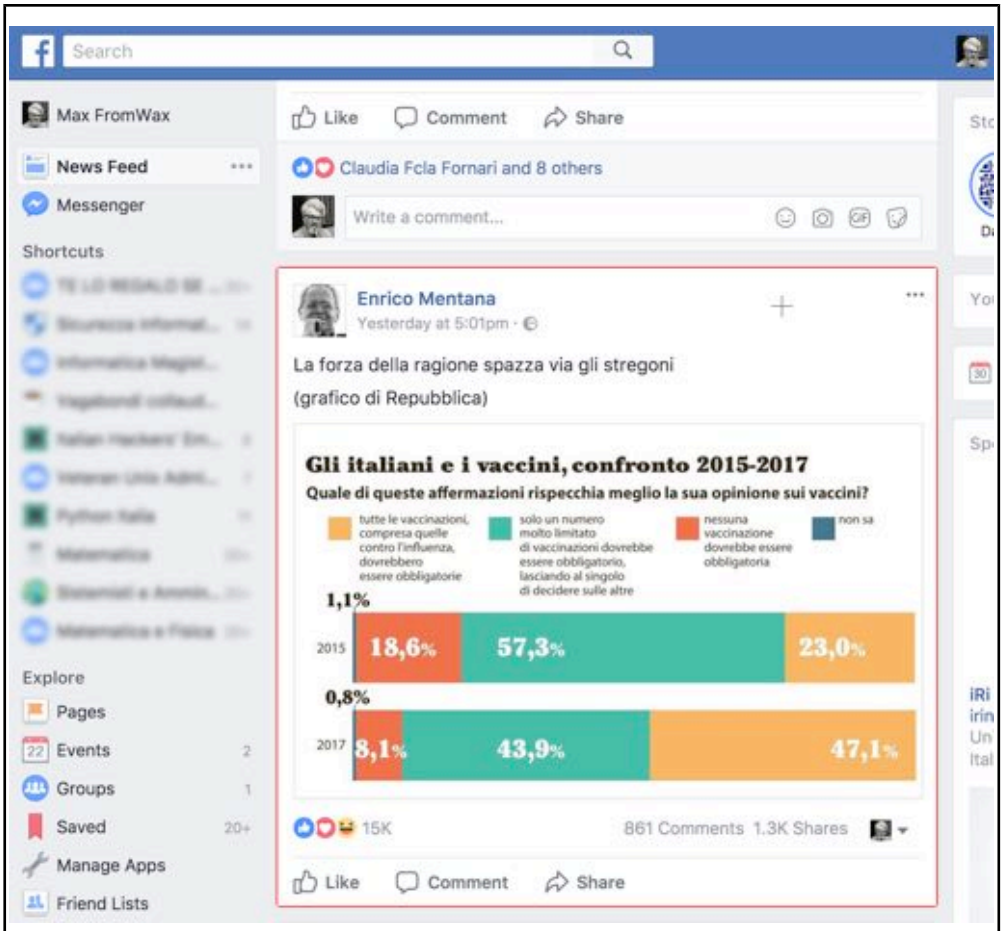


Figura 12: Webidence 1.0 - acquisizione post facebook

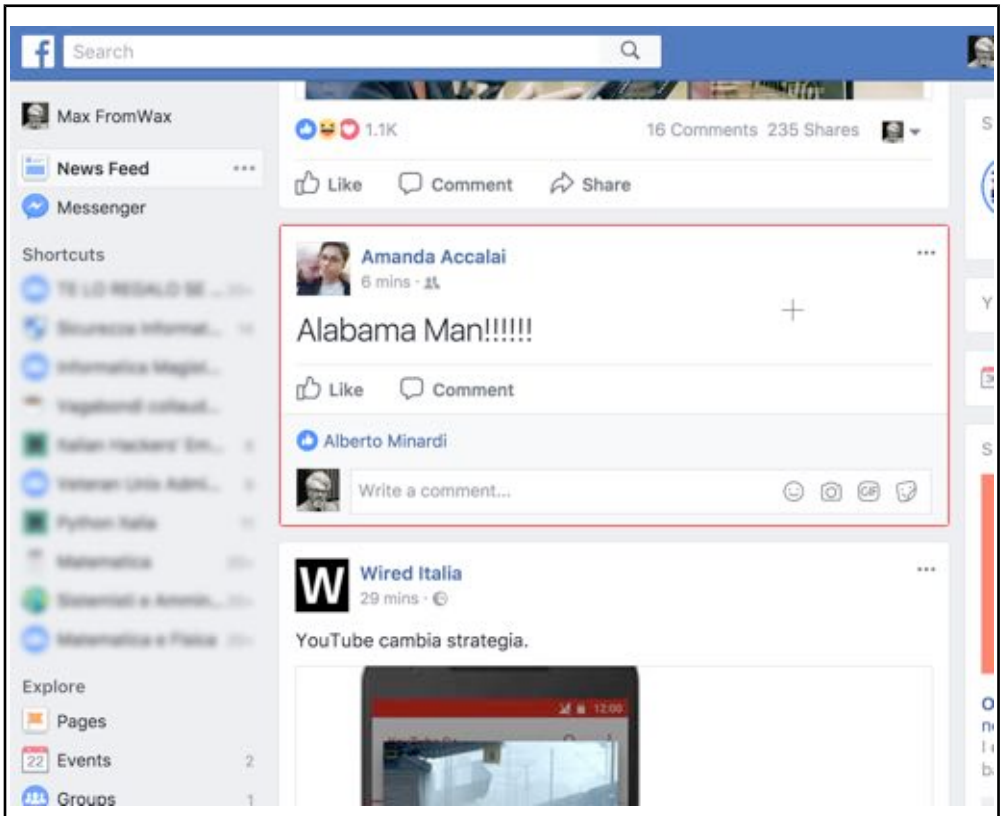


Figura 13: Webidence 1.0 - acquisizione post facebook

4.1.2 Implementazione e strumenti

La tecnologia server side adottata in questa prima versione per la realizzazione di un sistema atto all'interazione utente, che permettesse di rendere fruibile lo strumento di acquisizione tramite la metafora del "meta-browser annidato", è stata basata prevalentemente sull'uso di Python3 congiuntamente all'utilizzo del frame-work **Flask** (http://flask.pocoo.org/) e della libreria **Requests** (http://docs.python-requests.org/en/master/).

Flask

Flask viene definito dai suo stessi ideatori come un "*microframework*" (micro like as "micro-kernel), cioè una struttura "core" flessibile con minime funzionalità, concentrate nel definire determinate convezioni che aiutino a rendere l'ambiente omogeneo – con supporto al pattern MVC e alle estensioni – così da permettere di ampliare le potenzialità dell'ambiente di lavoro in base alle reali necessità. Inoltre offre un sistema di configurazione e di esecuzione centralizzato così da rendere l'ambiente facilmente gestibile con un approccio comune che sia trasversale a tutte le estensioni. Flask per esempio non ha un suo sistema ORM prefissato o un sistema di gestione di ruoli e utenze già integrato, ma sono aspetti che risultano facilmente aggiungibili in base alle proprie necessità e le relative extensions.

Risultano abbastanza ovvie le motivazioni che hanno spinto verso la scelta di Flask come framework su cui basare il progetto, in quanto si era alla ricerca di un prodotto che potesse offire un ragionevole compromesso tra avere dei solidi binari di partenza che permettessero di disporre di una struttura di riferimento e allo stesso tempo disporre di un'ampia libertà di scelta implementativa, il tutto senza aggiungere overhead di librerie che difficilmente sarebbero state utilizzate; per esempio, era stata anche valutata la possibilità di utilizzare **Django**, ma lo si è ritenuto troppo monolitico e troppo constringente, prevedendo che con esso si sarebbe corso il rischio di avere un prodotto che interferisse troppo nelle decisioni tecniche implementative che di volta in volta sarebbero potute emergere.

Requests

Per la parte ricoprente il ruolo di "mediatore" tra le richieste utente e il rilancio delle stesse verso il sito di interesse, si è ricorsi alla libreria **Requests** la quale annovera

interessanti e utili caratteristiche che permettono una gestione flessibile ed elegante delle richieste HTTP, fornendo quindi una solida base per l'implementazione del proxy di mediazione.

Tra le caratteristiche sicuramente più utili allo scopo del progetto si menzionano:

- Keep-Alive & Connection Pooling
- Sessions with Cookie Persistence
- Elegant Key/Value Cookies
- Multipart File Uploads
- Streaming Downloads
- Connection Timeouts
- Chunked Requests

In Python esiste nativamente nelle sue librerie standard il modulo urllib2 che mette a disposizione quasi tutte le principali funzionalità HTTP, ma la sua interfaccia è molto frastagliata. Quel modulo è stato creato per tempi diversi - e un web diverso. Serve molto lavoro (addirittura anche l'overriding di metodi) per realizzare il più semplice dei task. Requests si fa carico di tutto il lavoro per implementare HTTP/1.1 su Python - rendendo immediata l'integrazione delle applicazioni con i web services. Non c'è bisogno di aggiungere manualmente query string agli URL, o di fare form-encoding dei dati inviati via POST. Il Keep-alive e il pooling delle connessioni HTTP sono completamente automatici, tutto ciò grazie a urllib3, che è contenuta dentro Requests e permette ad ogni richiesta HTTP inviata all'interno di una sessione di usare automaticamente la stessa connessione, consentendo in tal modo di non avere problemi di compatibilità con servizi particolarmente pignoli (come per esempio Facebook o Twitter che tentano in tutti i modi di identificare sessione mediate così da evitare, per motivi di sicurezza, connessioni indirette che passani attraverso connessioni aperte da altre applicazioni, in grado quindi di tracciare le azioni degli utenti oltre che prendere conoscenza delle credenziali).

Di fondamentale importanza è stato il pieno e flessibile supporto nella libreria Requests ai cookies, la cui specifica implementazione ha permesso di creare agevolmente un componente software che facesse da mediatore in grado di mantenere viva la sessione generata dall'utente dal suo browser senza che questa venisse persa nella triangolazione.

Puntamento delle risorse

Nel momento in cui la connessione transita dalla web application intermedia verso l'utente, risultava necessario poter intervenire anche sulle risorse puntate internamente alla pagina (css,script,immagini,video,etc). Questo aspetto, seppure non dettato da nessuna necessità forense, aveva il "solo" scopo di soddisfare precise necessità tecniche, come evitare che la fruizione dei contenuti potesse essere contrastata dal server originario qualora avesse avuto da "obiettare" nel vedere la sessione dell'utente provenire da differenti indirizzi IP.

Si rendeva quindi necessario effettuare le opportune analisi all'interno del codice sorgente proveniente dal server di origine, così da poterne apportare le necessarie modifiche e far puntare coerentemente tutte le risorse verso un unico indirizzo intermedio. Potendo sfruttare gli stessi controlli di ispezione del codice, si rendeva inoltre possibile intercettare e modificare opportunamente quegli script, introdotti intenzionalmente dal fornitore del contenuto, con l'intento di impiantare "scomodi" controlli idonei ad intercettare potenziali anomalie per applicare relative protezioni utili allo scopo.

Purtroppo però, seppure innocue, le analisi e relative modifiche al codice originale non proponevano un panorama ottimale ad un interlocutore non tecnico (come per esempio un giudice o un avvocato), per il quale la presenza della parola "modifica" risuonava funestamente come una potenziale minaccia all'originalità (ed integrità) del contenuto. Proprio questo aspetto si è rivelato essere un punto particolarmente delicato, che ha contribuito ad interrompere lo sviluppo della fase 1.

Seppure le modifiche fossero solamente strumentali a rendere possibile la fruizione del contenuto all'operatore, senza che queste apportassero alcun impatto finale all'acquisizione, ad orecchie non tecniche, la sola presenza di potenziali "modifiche" risultava sollevare perplessità inaccettabili da far ricadere un alone negativo su tutta la soluzione.

Acquisizione dei contenuti

Per meglio ottimizzare le risorse, la fase di acquisizione era stata pensata affinché non partisse immediatamente al momento stesso del click di "finalizzazione". Si era notato infatti che certe acquisizioni risultavano essere così particolarmente esose di risorse da ricadere negativamente su tutta l'infrastruttura e di conseguenza su tutti gli utenti che stavano operando nel stesso momento sul medesimo asset. Si era quindi proceduto a calibrare le attività organizzando le singole acquisizioni in task "schedulati". Ogni

acquisizione veniva messa in cosa al momento del click dell'utente sul pulsante "acquisisci" (o finalizza). Così facendo si rincorreva l'obiettivo di bilanciare il carico in base ad un predeterminato numero massimo di acquisizioni parallele definito in fase di configurazione.

Una volta superata la coda e iniziato il vero processo di acquisizione l'url di interesse veniva dato in pasto a ***PhantomJS*** (http://phantomjs.org/), un browser non visivo "Full web stack". Seppure privo di rendering a video e con la sola possibilità di essere controllato da linea di comando o tramite scripting (javascript), sotto al cofano presenta un browser completo basato su Chromium con pieno supporto agli standard moderni del Web, potendo vantare un interprete javascript basato sul motore V8 di Google.

Insieme all'url di interesse si passavano a PhantomJS anche tutte quelle informazioni salvate durante la prima fase di navigazione svolta dall'utente, contenenti tutti gli header HTTP (quindi comprensivi anche dei cookie) necessari a replicare la sessione al momento dell'avvenuto click sul pulsante "finalizza". Quindi se l'utente aveva effettuato un login su un determinato servizio, tale sessione risultava disponibile ugualmente in fase di acquisizione, anche a fronte di controlli puntuali sull'indirizzo IP di provenienza da parte del servente. Essendo infatti che tutta la navigazione risultava mediata, anche l'IP di uscita sarebbe risultato mascherato, presentandosi quindi con il medesimo indirizzo. Discorso simile per lo user-agent, il quale essendo un header HTTP poteva essere tranquillamente clonato, rimanendo così invariato anche nella fase finale di acquisizione.

In conclusione, una volta giunti con PhantomJS configurato in maniera tale da replicare la sessione utente e l'url della risorsa da acquisire, si applicavano svariate strategie tali da permettere a PhantomJS di compiere quelle azioni necessarie ad ottenere il contenuto completo. Per fare ciò si erano studiate azioni "standard" da compiere ad ogni accesso ad un contenuto, così da avere una ragionevole certezza che le risorse in essa presenti fossero completamente caricate. Esempio di azione compiuta ad ogni accesso era lo "scrolling" della pagina fino a che nessun nuovo contenuto venisse aggiunto dinamicamente, oppure passare il "mouse virtuale" su ogni oggetto del DOM, così da permettere il caricamento di tutte quelle risorse secondarie caricate allo scatenarsi dell'evento "onmouseover". Tutti comportamenti, seppure non privi di mancanze ed effetti collaterali difficili da arginare, utili allo scopo di determinare quelle risorse che, per ottimizzazione dello sviluppatore, non erano caricate insieme alla pagina.

Una volta svolte tutte le azioni di base necessarie, alcune delle quali studiate appositamente per pre-determinati domini noti possedere caratteristiche peculiari, si procedeva quindi al download di tutte le risorse caricate dalla pagina. Questo aspetto risultava essere possibile grazie al supporto nativo al "*Network Monitoring*" integrato in PhantomJS (http://phantomjs.org/network-monitoring.html) caratterizzato dalla possibilità di creare e associare specifiche callback richiamate al momento del caricamento della risorsa, potendo così generare al termine del caricamento un export in formato HAR (HTTP Archive) (http://www.softwareishard.com/blog/har-12-spec/) facilmente fruibile con HAR viewer (http://www.softwareishard.com/blog/har-viewer/) (Figura 14).

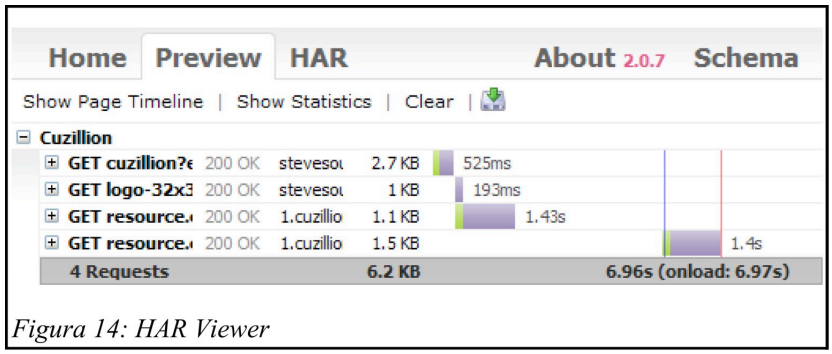


Figura 14: HAR Viewer

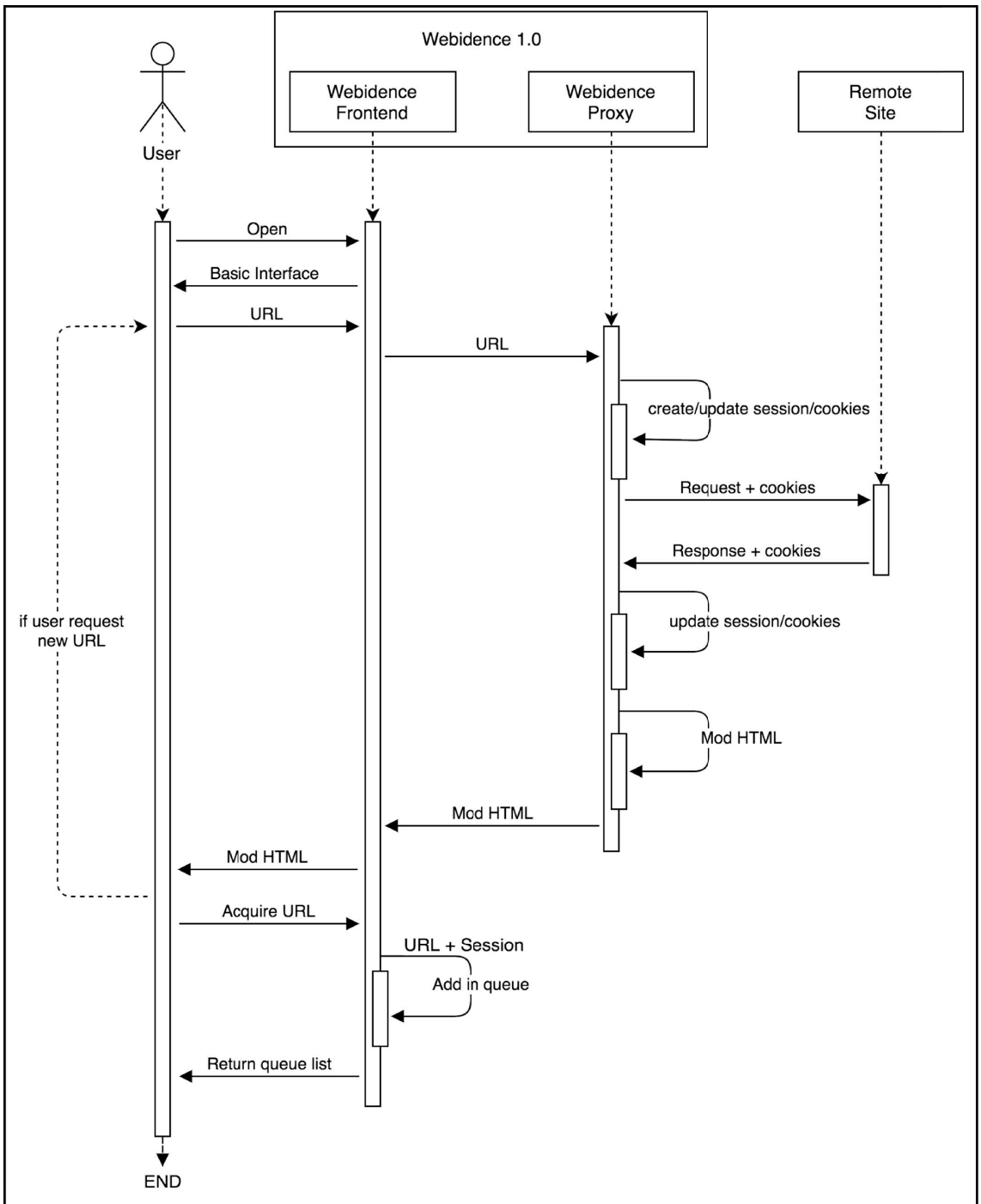


Figura 15: Webidence 1.0: Sequenze Diagram "acquisizione"

4.1.3 Problemi riscontrati

Il progetto così come appena illustrato poteva avere un funzionamento ragionevolmente accettabile se l'obiettivo finale non fosse stato un sistema di acquisizione pensato per vivere in un contesto forense. I limiti intrinsechi, sia tecnici che giuridici, ne hanno quindi decretato l'abbandono portando come conseguenza alla realizzazione di un nuovo progetto basato completamente su differenti tecnologie, sempre con l'intento di mantenere per la metafora del "meta-browser annidato".

Di seguito saranno illustrate le singole criticità individuate e per ognuna se ne discuteranno i limiti riscontrati, così da poter dare una panoramica ponderata e le relative motivazioni che hanno contribuito al cambio radicale di progetto.

Indeterminatezza

Una cosa molto importante in un'acquisizione è la capacità di raccogliere puntualmente quanto realmente di interesse e non anche elementi estranei.

Consistendo l'interfaccia utente in una navigazione indiretta, nata con l'obiettivo di far giungere l'utente alla risorsa di interesse che dovrà poi essere acquisita, proprio questa caratteristica genera essa stessa un problema di "indeterminatezza". Il problema manifestato introduceva il rischio concreto di rendere incerta la fase di acquisizione, non garantendo che il contenuto da acquisire sarebbe risultato essere proprio quello che l'operatore si trovava davanti agli occhi al momento dell'accodamento del task di acquisizione. Ma cosa ancora peggiore, oltre alla non pedissequa somiglianza non si poteva neppure garantire la stessa presenza del dato di interesse; si pensi per esempio ad un commento su un forum, il quale sarebbe potuto slittare in pagine successive a favore dei contenuti più "freschi", oppure si consideri quei contenuti mostrati a fronte di specifiche politiche basate su fattori non aleatori : commenti in una "top 10" generati in base ai voti degli utenti, advertising presenti nelle pagine (come gli AD sense di Google) che cambiano con fattori decisionali semi-random.

A fronte di una prima e superficiale analisi ci si potrebbe aspettare che il problema di ottenere un output non conforme a quanto inizialmente visionato sia indotto dalla sola schedulazione dell'attività, la quale rinvia l'accesso al dato anche se solo di pochi minuti; Problema che si manifesta palesemente in tutte quelle pagine dove i contenuti sono aggiornati frequentemente oppure dove l'ordine proposto risulti pseudo-casuale.

Un esempio ben noto è la timeline di facebook, la quale cambia, spesso anche radicalmente, da un refresh al successivo, oppure una ricerca effettuata sul motore di ricerca Google, la quale può dare risultati non sempre uguali anche a distanza di pochi minuti, sopratutto per quanto riguarda la parte di advertising che potrebbe proprio essere la parte incriminata di interesse.

Questo problema di "indeterminatezza" sopraggiunge invece anche se l'acquisizione non viene schedulata, ma immediatamente avviata al momento stesso dell'invocazione e ciò in quanto il problema di indeterminatezza è intrinseco alla generazione di una nuova richiesta HTTP. Tutto questo a prescindere che essa sia immediata o a distanze temporali arbitrariamente corte, rendendo quindi necessaria una fase di acquisizione a "tempo zero" in grado di rappresentare esattamente quello che l'operatore si ritrovi in quello specifico momento stampato a schermo e che non richieda di effettuare nessuna nuova richiesta HTTP che inevitabilmente ripresenterebbe il medesimo rischio.

A fronte di questa problematica, ancor prima di abdicare il progetto in favore di uno nuovo, si era anche ragionato sulla possibilità di cambiare la fase di acquisizione, basandola non più su una nuova richiesta svolta da zero in base all'url richiesto dall'operatore, ma elaborandola partendo dall'ultimo codice HTML proposto al client prima della modifica effettuata dal proxy, ma anche questa soluzione non avrebbe portato a risultati idonei, in quanto il codice di base sarebbe stato un codice troppo grezzo e privo delle modifiche indotte dall'attività client side; si pensi per esempio ad un contenuto ottenuto dopo un ipotetico click e relativa chiamata asincrona che avesse avuto occasione di aggiornare il contenuto della pagina sul client dell'utente e di riflesso sul client "speculare" residente sul server.

Alterazione dei sorgenti

Quello che si andrà ad evidenziare non è un reale problema tecnico, quanto più un problema di "sentimento psicologico" indotto dalla strategia adottata per rendere fruibile nelle condizioni più idonee il contenuto richiesto dall'utente.

Come spiegato nei capitoli precedenti, a fronte di una richiesta utente di un determinato URL, la richiesta veniva raccolta e poi mediata da un agente software intermedio che, una volta ottenuta dal servizio remoto la risposta con il relativo codice HTML, ne modificava i riferimenti interni per farli indirizzare puntualmente al servizio intermedio, in modo tale che potesse proseguire a compiere la sua opera di "mediazione" anche per tutte le successive risorse statiche interne alla pagina. La mediazione così applicata

rendeva possibile veicolare tutto il flusso dati da un unico punto di uscita, evitando che remotamente si intercettasse del traffico asincrono anomalo in grado di far scattare blocchi sul server remoto.

Questa fase di "fetch & modify", effettuata dal componente intermedio, era solo preposta a rendere fruibile il contenuto all'interno del "meta-browser" lato utente e non avrebbe influenzato per nulla la fase di acquisizione. Anche se tecnicamente si potevano fornire tutte le garanzie possibili sulla fase di acquisizione, quando è stato mostrato il funzionamento dell'infrastruttura a personale non tecnico, ma comunque inserito nella realtà contestuale (si pensi per esempio ai giudici, agli avvocati, PM, alle forze dell'ordine, etc.), è stata più volte sollevata la critica in merito alla strategia di "fetch & modify" adottata che, seppure indolore e non influente ai fini dell'acquisizione, riusciva ad insinuare comunque delle perplessità; le stesse perplessità che un domani potrebbero essere sollevate durante un ipotetico procedimento legale da chiunque voglia obiettare sulla soluzione.

A fronte di queste critiche, seppur discutibili a livello tecnico, non si poteva fare a meno di rivedere questo aspetto. Purtroppo (o per fortuna) un prodotto deve vivere in un determinato contesto e, per ragioni "di mercato", non si possono ignorare le critiche sollevate dagli addetti ai lavori anche se esse non sono strettamente tecniche, sopratutto considerando che il prodotto dovrà essere successivamente utilizzato proprio da quelle figure professionali che hanno sollevato tali critiche; risulta quindi fondamentale (sopratutto in contesti come quello in esame) ottenere piena fiducia senza introdurre il minimo dubbio su quanto dovrà essere utilizzato. Essere troppo rigorosi nel difendere le proprie scelte tecniche, ignorando certe critiche e/o richieste, avrebbe come conseguenza la fuoriuscita dal mercato, pertanto è meglio non fare finta di nulla e cercare di migliorare l'approccio utente anche da un punto di vista "psicologico".

Video capture

Come nel caso precedente, ci sono state occasioni di confronto con personale non tecnico, ma particolarmente sensibile a certe tematiche e che potevano dare un loro contributo dall'esterno (giudici, avvocati forze dell'ordine, …), ed è stato sollevato un altra questione, ovvero la mancanza di un video che mostri cosa ha fatto e come l'utente è pervenuto al dato acquisito.

Anche in questo caso sarebbe superfluo affermare che si potrebbe ricostruire questa informazione fornendo i dettagli e il tracciato delle richieste svolte dall'utente e, in

maniera ancora più fine, le coordinate dei movimenti del mouse e i relativi click. Purtroppo queste informazioni in formato grezzo, seppur dettagliate e tecnicamente sufficienti, non darebbero nessun contributo a chi non ha padronanza con certe tematiche tecniche. Anche la possibilità di redirigere le varie coordinate in un video generato successivamente instillerebbe il dubbio dell'artificialità in un argomento tecnico che viene già visto molto astratto e artificiale (virtuale) di suo senza aggiungere ulteriori complicazioni.

Purtroppo lo strumento, così com'è nato, non poteva rendere possibile la generazione di una registrazione video dello schermo dell'utente che tracciasse in un filmato le azioni da lui compiute a meno di non richiedere l'installazione di un software dedicato a tale scopo; in tal modo però si sarebbe andati troppo lontani dal goal prefissato dal progetto, ovvero realizzare un prodotto finale agnostico dal sistema, che non fosse invadente negli ambienti operativi degli utenti, pertanto questa strada è stata totalmente scartata spostando questo task a livello server side, con quindi la necessità di rivedere nelle fondamenta il progetto.

Traffico di rete

Un limite non da poco, che peraltro veniva superato dalla "concorrenza" (LegalEye e FAW) era la mancanza della cattura del traffico di rete necessario per assicurare (seppure dentro a certi limiti) che le richieste inviate e ricevute fossero state indirizzate verso i giusti indirizzi.

Risultava quindi necessario non tracciare solamente il traffico rivolto verso le porte HTTP/S, ma anche tutte le richieste che potessero in un qualche modo far parte del traffico di rete intercorso dal momento della richiesta al momento della response; importante quindi era tenere traccia anche di tutte le richieste DNS che componevano tutta l'attività dell'utente, oltre che tutto il traffico di rete "sotterraneo" che ne identificasse la sobrietà delle connessioni, così da indicare l'isolamento del sistema stesso senza la presenza di traffico anomalo nelle attività dell'utente.

Anche in questo caso, a causa dell'infrastruttura del progetto, non era immediato poter estrapolare queste informazioni senza problematiche collaterali, il che portava il tutto ad essere parzialmente inadeguato rispetto alle necessità che gli obiettivi richiedevano. Anche per tale ragione, di conseguenza, la riscrittura dell'architettura del progetto si è vista necessaria.

Componenti client side

Ulteriore, ma non meno rilevante, il problema rappresentato dalla non piena estraneità del prodotto da componenti di terze parti che potevano essere presenti sulle macchine degli utenti, come per esempio anti-virus, ad-blocker, firewall (fisici o software), add-ons varie, … etc.

Questa circostanza rendeva particolarmente problematiche tutte quelle situazioni nelle quali un determinato contenuto poteva essere modificato, bloccato o addirittura eliminato dalla pagina a causa delle configurazioni del singolo sistema. In fin dei conti al client arrivava del codice HTML, che poteva essere riconosciuto come "sospetto" e attivare quindi delle azioni estranee non prevedibili.

Anche in questo caso, non potendoci imporre sul sistema degli utenti, diveniva necessario trovare una soluzione che non fosse influenzata da agenti di terze parti e per fare ciò l'unica possibilità era quella di impedire eventuali interferenze sulla componente di interfaccia, ma ciò risultava incompatibile con la struttura del progetto e ne prevedeva quindi un ripensamento globale.

Complessità e sostenibilità

Ultima problematica da evidenziare, presente sin dall'inizio dello sviluppo del progetto, è quella relativa alla gestione delle varie complessità che certe pagine propongono.

Quando si parla di "complessità", si vuole sottolineare come certe pagine del Web moderno veicolano i propri contenuti, spesso facendo uso di tante richieste asincrone, o usufruendo delle "*Web Socket*", o introducendo lato client side dei controlli javascript che tracciano le attività dell'utente frapponendosi tra le azioni di quest'ultimo.

Un esempio fra tutti è Facebook, il quale presenta un box trasparente al di sopra di tutta la sua interfaccia che traccia tutte le azioni compiute dal cursore dell'utente, tant'è che ogni click effettuato non attiva la normale azione del browser quando si clicca su un elemento attivo, ma viene intercettato e poi passato all'hook di riferimento che ne blocca l'azione di default e lo elabora separatamente per poter così svolgere un'azione asincrona che caricherà solo la parte dell'interfaccia necessaria a mostrare il contenuto richiesto, evitando in questo modo il ricaricare completo della pagina. Questo aspetto ha introdotto non pochi problemi e ciò in quanto, seppure in un browser normale la URL-bar cambi, questa azione non veniva però recepita a livello di "meta-browser", con il conseguente problema di non ottenere il cambio di pagina o non ottenere l'indirizzo

giusto da acquisire. La circostanza appena illustrata ha comportato il dover scrivere codice javascript da iniettare puntualmente nel codice HTML risultante ove si rendeva necessario e, poiché queste casistiche sono ognuna diversa dall'altra, si sono dovuti implementare differenti codici per differenti siti (Facebook, Twitter, Instagram, etc.).

Questo in parte si ricollega alla problematica della "modifica", nel senso di alterazione del dato originale, ma anche a "nuovi" problemi prettamente tecnico/economici, poiché studiare caso per caso ha dei costi alti di implementazione e ancora più impattante è la continua necessità di monitorare il funzionamento del prodotto nel tempo, considerando che gli aggiornamenti possono essere all'ordine del giorno. Ad esempio, nel caso di alcuni servizi (Facebook), la modifica di certe componenti avviene anche più volte al giorno e capita spesso che ciò comprometta la compatibilità del prodotto oggetto di sviluppo con conseguente disservizio causato agli utenti, poiché per risolvere certe casistiche è necessario, oltre alla prontezza di rilevamento del problema, anche il tempo tecnico di sviluppo per rimanere al passo.

Questa situazione, seppure in un qualche modo aggirabile con tecniche varie che contrastino determinate circostanze indotte dal codice (volutamente) introdotto da certi servizi, ha avuto un peso molto determinante nel rivalutare la sostenibilità del progetto, in quanto oltre alla non sempre piena compatibilità con "tutto" quello che offre il Web, sarebbe richiesto anche molto lavoro per mantenere la (ridotta) compatibilità acquisita con grandi sforzi, portando di conseguenza ad investimenti tecnico/umani non indifferenti che sono stati valutati come **insostenibili**.

Gestione certificati SSL/TLS per traffico HTTPS

Questo aspetto non era presente nella fase 1.0 in quanto non era stata considerata la fase di acquisizione del traffico di rete e quindi non vi era neppure la necessità di decifrare il traffico cifrato acquisito. Ma, come meglio si andrà ad illustrare nei capitoli successivi dedicati a tale argomento, vedremo come ciò abbia avuto una rilevanza molto importante proprio per via dell'introduzione dell'acquisizione del traffico di rete generato.

4.1.4 Conclusioni

Seppure a fronte di tanto lavoro e studio, che comunque sono tornati utili per la fase successiva, ci si è visti costretti ad abbandonare la strada a favore di una che portasse il

progetto in una direzione nuova, più affidabile per gli operatori, più compatibile con il complesso mondo del Web e economicamente sostenibile in ottica aziendale.

Forse alcune problematiche legate alla complessità sarebbero anche potute essere risolte, con più tempo e studio sulle singole casistiche, ma sarebbero comunque rimasti molti dei vincoli sopra elencati, sia forensi che economici. Inoltre, non volendo essere un progetto "sviluppa e getta" che vivrà solo per il periodo della presente tesi, ma un progetto che dovrà avere una vita produttiva in una realtà aziendale vera, non si poteva (e non si può) ignorare anche i costi derivati da una determinata implementazione e, seppure con grande rammarico tecnico, si è dovuto per forza tagliare linfa vitale alla "fase 1" per trovare un nuova strada sostenibile sia economicamente ma anche in ottica forense.

Successivamente verrà illustrato il progetto finale indicato come "Fase 2" o "Webidence 2.0", ma era importante mostrare la "Fase 1", in quanto un risultato negativo in realtà è un risultato positivo, poiché indica e mette in evidenza delle criticità importanti e che danno quindi una giusta prospettiva oltre che motivazioni forti utili al progetto successivo. Illustrare la "Fase 1" è stato quindi importante proprio in questa ottica, in quanto il lavoro svolto, seppure arrivato ad un punto morto, ha dato comunque i suoi frutti proponendo uno studio di "non fattibilità" che ben giustificherà l'approccio successivo.

4.2 Fase 2 (Webidence 2.0)

4.2.1 Panoramica

Questa seconda fase, denominata anche “Webidence 2.0”, è la diretta evoluzione indotta dal fallimento della “fase 1” e riesce a sopperire alle mancanze di quest'ultima. Questa seconda evoluzione, pur facendo uso di approcci e tecniche molto differenti, tiene invariata la metafora di utilizzo del “meta-browser annidato” vista per la fase 1.

In questa nuova versione il meta-browser non è più un interfaccia HTML pura che fa da wrapper al contenuto Web fruito dal server e offerto all'utente, ma è invece un vero e proprio browser sviluppato per l'occasione, che vive in un processo in esecuzione sul server e viene renderizzato remotamente, con un’interfaccia poi mostrata sul browser dell'utente tramite uno streaming continuo e un interscambio continuo degli eventi scatenati dall'utente che vengono replicati in “real-time” sul server applicativo.

Per alcuni aspetti, questo tipo di approccio può essere assimilabile alla tecnologia adottata da LegalEye, ma nel caso del presente progetto il sistema non è basato su una macchina virtuale che mette in esecuzione un desktop completo, bensì su un container che esegue un browser (più relative utility che saranno spiegate in seguito) in un processo isolato e invia il rendering remoto al client, veicolando il tutto tramite un'interfaccia HTML5.

Ulteriore differenza rispetto a LegalEye è il browser utilizzato, che non è un browser generico pre-esistente, ma uno sviluppato da zero con il preciso intento di essere “forensically friendly”, che possa quindi aiutare l'utente nello svolgimento del proprio task di acquisizione senza portare a banali errori operativi. Proprio tale approccio va ad accomunare la seconda fase a quello adottato dal prodotto FAW, con il vantaggio di rendere il browser un'entità che vive remotamente in una sandbox controllata da terze parti che ne garantiscono l'affidabilità e l'imperturbabilità dell'ambiente ospitante.

Risulta quindi immediatamente evidente da questa breve panoramica come la seconda fase cerchi di prendere il meglio delle tre soluzioni appena illustrate, Fase1, LegalEye e FAW, così da portare e integrare tutte le caratteristiche reputate interessanti sotto il cappello di un unico prodotto. Così facendo si persegue l'intento di colmare le lacune che ogni prodotto si porta dietro per generare un unico strumento che ne sia privo.

Scopo di Webidence 2.0 sarà quindi quella di implementare le seguenti caratteristiche:

- utilizzo senza la necessità di installare nulla sulla postazione dell'operatore
- implementazione della metafora del meta-browser annidato
- integerrima fedeltà al codice visionato senza apportare mai nessuna modifica
- acquisizione del codice RAW e del codice post-elaborazione client
- registrazione della sessione video di tutto il processo di utilizzo fino al termine dell'acquisizione
- estrazione di tutto il traffico di rete intercorso
- capacità di effettuare lo screenshot delle pagine
- superamento del non determinismo del dato, in quanto sarà il forensics-browser a catturare il codice nell'attimo prescelto dall'utente
- superamento di ogni complessità indotta dal fornitore del contenuto, essendo tutto gestito da un vero browser
- totale tracciabilità delle azioni utente, in quanto oltre all'attività compiute dal cursore sarà posizionata una a virtual keyboard che mostrerà a video i tasti realmente digitati

4.2.2 Utilizzo

L'utilizzo ad alto livello non è molto dissimile dalla precedente implementazione, si è voluto fortemente mantenere lo stesso approccio del meta-browser adottato nella fase 1, dato che lo si è reputato il più adatto e semplice per lo scopo prefissato.

In generale, mantenere una coerenza di utilizzo con quanto già a conoscenza dell'utente risulta una buona strategia poiché aiuterà l'utilizzatore ad avere una curva di apprendimento che risulti la meno ripida possibile, così da rendere la fruizione del servizio più proficua e soddisfacente. Questo aspetto risulterà non secondario anche in fase di perizia tecnica, quando sarà necessario spiegare le procedure adottate a lettori che, con elevata probabilità, non saranno avvezzi a tecnicismi e poco apprezzerebbero ritrovarsi davanti soluzioni “aliene”. Risulterà quindi strategicamente importante poter usare un vocabolario comune che non dia adito a dubbi o malsane e libere interpretazioni, tali da mettere in cattiva luce la prova generata.

L'utilizzo, come già ampiamente descritto per la fase 1, sarà tutto veicolato da un normale browser e la fase di acquisizione potrà essere compiuta tramite un “semplice” click, che alla sua attuazione metterà insieme tutti i pezzi necessari alla realizzazione di un archivio finale che conterrà:

1. il codice finale della pagina acquisita,
2. il codice raw della prima richiesta HTTP,
3. tutte le risorse (immagini, video, css, js, …) legate alla pagina,
4. per ogni risorsa scaricata al punto 3 e i codici HTML di cui al punto 1 e 2, saranno tenuti traccia le response HTTP e gli eventuali redirect che hanno portato alla risorsa,
5. il video della sessione di acquisizione, che comprenderà quindi tutta la fase di navigazione precedente che ha portato all'individuazione della risorsa acquisita,
6. il dump completo del traffico di rete di tutta la sessione (compatibile con WireShark),
7. le master key SSL/TLS adottate dal browser, necessarie a decifrare il dump di rete in merito alle richieste veicolate tramite protocollo HTTPS,
8. un report finale **marcato temporalmente** che descriverà l'attività svolta e gli strumenti utilizzati, oltre a riportare gli hash di tutte le risorse presenti in archivio e i relativi riferimenti temporali

Di seguito si darà una panoramica dell'utilizzo raccontandone la storia di interazione.

Fruizione come servizio

L'utente si collega al portale Web del servizio, si registra con fornendo le proprie generalità e una volta approvata la sua iscrizione potrà accedere alla propria dashboard facendo uso delle credenziali assegnategli.

Una volta acceduto alla propria dashboard, potrà visionare le acquisizioni svolte oppure decidere di avviarne una nuova se il credito a disposizione lo permette.

Nel caso in cui l'utente decida di procedere con una nuova acquisizione, si aprirà una pagina con al suo interno annidata una seconda pagina contenente l'interfaccia del

browser forense, con precaricata una land-page nella quale si fornirà una panoramica che potrà aiutare l'utente ad orientarsi con lo strumento.

L'interfaccia del meta-browser è stata pensata volutamente per essere la più minimale possibile, rimuovendo ogni componente superfluo che potesse contribuire a creare confusione a chi dovrà fruire della prova generata la quale conterrà, si ricorda, sia codice sorgente che le risorse del contenuto Web di interesse, oltre che la registrazione video completa della sessione dell'attività e tutto il traffico di rete generato. Sarà quindi importante generare un contenuto fruibile nella maniera più lineare possibile, impedendo all'utente ogni genere di utilizzo che possa generare confusione in chi effettuerà la visione del video o analizzerà il traffico di rete. A tal proposito sono stati rimossi i tasti con le opzioni classiche di un browser, tra cui "l'avanti e indietro", così da obbligare l'operatore ad andare solamente avanti o eventualmente poter esplicitare un nuovo indirizzo nella barra di navigazione seguito da un chiaro click sul pulsante "vai".

Altra caratteristica introdotta è l'impossibilità di utilizzare la tastiera fisica del proprio computer, rimandando tale funzionalità alla tastiera virtuale attivabile dal relativo pulsante presente sulla barra di navigazione del meta-browser. Questo accorgimento è stato apportato in quanto, se è possibile registrare un video della sessione di quanto avviene nello schermo virtuale, non è possibile registrare ciò che avviene dall'altra parte dello schermo e, più precisamente, le mani che operano sulla tastiera. L'introduzione della tastiera virtuale ha quindi l'obiettivo di rendere l'attività di acquisizione la più esplicita possibile durante la fruizione del video da parte di terzi soggetti, che avranno così modo di vedere in maniera chiara ed evidente anche quali tasti sono stati premuti sulla tastiera, per dissolvere qualsiasi ragionevole dubbio sulla possibilità di attribuire all'operatore la composizione di qualche esotica combinazione della tastiera che possa aver apportato alterazioni nel contenuto.

Giunti a questo punto, risulta abbastanza chiaro che l'utente avrà di fronte un sistema che in fase di acquisizione gli permetterà di effettuare una navigazione relativamente normale e forzatamente guidata a generare un contenuto il più lineare possibile. Una volta che l'operatore sia giunto al contenuto Web di interesse non dovrà fare altro che finalizzare il tutto spingendo il solo pulsante con sopra scritta l'etichetta "finalizza" che, a quel punto, darà il via al salvataggio sia del codice sorgente "raw" del contenuto che del codice sorgente puntuale della rappresentazione del DOM in quel momento visualizzato dal meta-browser, e il tutto sarà completato delle singole risorse statiche richieste. Una volta acquisiti i contenuti di interesse il tutto sarà cristallizzato insieme ai

log HTTP di tutti i contenuti richiesti dal browser, alla registrazione video della sessione e al traffico di rete generato dall'inizio alla fine della sessione. In conclusione, per ogni file prodotto sarà generato un hash SHA1 che ne garantisca l'integrità.

Fruizione come sistema live

Come modalità di fruizione si è valutata anche la possibilità di non offrire il prodotto come servizio ospitato su server di terze parti, ma di distribuirlo sotto forma di ISO "live" che ogni utente potrà autonomamente avviare su di un proprio sistema.

La fruizione tramite sistema "live" avrebbe il vantaggio di non pretendere da parte dell'utente la predisposizione di un sistema e l'investimento di tempo per le relative configurazioni, ma gli pemetterebbe di usufruire di un sistema già pensato per essere "pronto all'uso", da avviare o su una macchina fisica o una macchina virtuale, potendo quindi creare all'occorrenza delle workstation atte all'acquisizione di contenuti Web.

La dinamica di utilizzo non sarà molto differente da quella precedentemente illustrata, se non che l'utente dovrà collegarsi all'indirizzo IP della macchina e non avrà necessità di registrarsi in quanto la dashboard sarà fruibile senza credenziali specifiche.

4.2.3 Problemi risolti

Come già precedentemente indicato, la fase 2 si è resa necessaria per poter arginare i problemi insorti nella fase 1, illustrati nel capitolo 4.1.3.

Si esporrànno di seguito, in modo dettagliato, i singoli punti critici di Webidence 1.0 precedentemente illustrati e come i nuovi approcci adottati da Webidence 2.0 abbiano risolto ognuno di essi.

Indeterminatezza

Se in Webidence 1.0 il contenuto visionato a schermo in fase di attività poteva non coincidere perfettamente con quanto successivamente il sistema andava in effetti ad acquisire, questo non accade nella versione 2.0 in quanto l'acquisizione risulta già essere attiva in background nel momento stesso in cui l'operatore effettua la visione dei contenuti che in conclusione andrà a finalizzare. Questo accorgimento rende di conseguenza ogni contenuto puntualmente omogeneo con quello effettivamente visionato e del quale si richiede l'acquisizione.

Alterazione dei sorgenti

In Webidence 1.0 era necessario introdurre delle alterazioni nel codice della pagina per forzare le singole risorse a puntare verso i server ospitanti che svolgevano da proxy intermedio. In Webidence 2.0 non vi è più la necessità di introdurre codice nelle pagine fruite in quanto tutto viene realmente visionato all'interno di un browser eseguito in un processo indipendente che risiede sul server, e il rendering viene inviato al browser dell'utente con uno streaming costantemente.

Video capture

Non essendo (ad oggi) possibile ottenere una condivisione dello schermo in maniera nativa su tutti i browser, presentandosi in alcuni casi la necessità di installare plugin aggiuntivi, risultava arduo (se non impossibile) creare una registrazione video della sessione di acquisizione che documentasse la "strada" e le azioni compiute dall'utente per raggiungere la risorsa di interesse.

Inoltre, sia che si facesse uso del supporto nativo a "WebRTC Sharing Screen" che tramite plugin aggiuntivi, risultava comunque necessaria la corretta azione di scelta da parte dell'utente. Proprio questa richiesta di interazione avrebbe potuto introdurre errori operativi, con il relativo rischio di rendere inutile la prova video.

Poiché lo scopo del progetto è anche quello di creare uno strumento che guidi l'operatore verso una corretta acquisizione in modo agevole, completo e sicuro, la presenza di una scelta così delicata e non gestibile si è ritenuto che fosse troppo "delicata" per essere lasciata nelle mani dell'utente.

Webidence 2.0 evita il presentarsi di questo ostacolo, in quanto tutto il processo esecutivo del meta-browser risulta essere residente sul server e così anche lo schermo virtuale dove esso viene renderizzato. Risulterà quindi sufficiente attuare una registrazione video di tale schermo virtuale per ottenere tutta l'attività di acquisizione correttamente registrata in un ottimale spazio asettico, privo di "perturbazioni" che potrebbero introdurre dannose imperfezioni.

Traffico di rete

Essendo Webidence 2.0 un sistema in esecuzione come processo autonomo su di un sistema server ospitante, risulta immediato risolvere il problema dell'intercettazione del

traffico di rete, in quanto ogni esecuzione genera un proprio traffico ben distinto e discriminabile facendo uso dello "User ID" (UID) con cui il processo è stato eseguito.

In Webidence 1.0 questo aspetto non poteva essere realizzato in quanto tutta l'attività era separata su più punti:

- traffico generato dal client utente,
- traffico generato dal server durante la fase interattiva ed essa stessa suddivisa ulteriormente in due direzioni, una dal server verso il Web e la seconda dal server verso il client,
- traffico generato dal server durante la fase di acquisizione non interattiva

Risultava sia difficile da implementare sia impossibile, oltre che inutile ai fini probatori:

- **difficile**, in quanto richiedeva la necessità di suddividere il tracciamento del traffico nei 3 spezzoni distinti poc'anzi esposti
- **impossibile**, in quanto il traffico realmente generato dal client utente non poteva essere totalmente messo sotto tracciamento, ma se ne poteva dare solo un estratto parziale
- **inutile**, in quanto l'attività di acquisizione era suddivisa in 2 fasi distinte:
  1. l'operatore naviga tramite il meta-browser verso il contenuto di interesse e ne richiede l'esecuzione
  2. il server mette in coda l'acquisizione che verrà effettivamente messa in esecuzione successivamente

  Pertanto, tale suddivisione dell'attività comportava il rischio concreto che il tracciamento del traffico non avrebbe dato un risultato omogeneo, in grado di essere utili ai fini probatori.

Componenti client side

Il problema derivante dagli effetti collaterali causabili da componenti di terze parti installati sulla macchia client dell'operatore è per definizione aggirato dall'architettura adottata da Webidence 2.0. Se in Webidence 1.0 la fruizione del contenuto, in quanto formato da componenti HTML, poteva essere influenzata e/o bloccata da potenziali agenti software di sicurezza (antivirus, web filtering, ad-blocker, etc) residenti sul sistema dell'operatore, in Webidence 2.0 questo aspetto è del tutto arginato in quanto il

contenuto fruito dal client non comprende le reali componenti originali del sito remoto, ma la fruizione avviene tramite streaming binario e quindi, anche se il contenuto veicolato violasse le policy del sistema ospite, ciò non risulterebbe identificabile.

Complessità e sostenibilità

In Webidence 1.0 sono stati rilevati non pochi problemi per ottenere la piena (ma anche solo parziale) compatibilità verso determinati servizi comprendenti tecnologie poco amichevoli, se non proprio ostili, verso chiunque tentasse di frapporsi alla fonte originale.

Anche in questo caso, il problema è automaticamente arginato dalla fruizione tramite un vero browser (seppur limitato) senza nessuna mediazione intermedia che possa sollevare le eccezioni di quei servizi particolarmente pignoli.

Questo aspetto rende quindi anche lo sviluppo decisamente più sostenibile, sia qualitativamente che quantitativamente, permettendo di orientarsi verso il goal primario di generare uno strumento forense e non un sistema di "evasion" di navigazione.

Gestione certificati SSL/TLS per traffico HTTPS

Avendo la possibilità di gestire in autonomia la parte applicativa del meta-browser risulta potenzialmente possibile salvare la master key e la client random key TLS per poter permettere successivamente la decifrazione completa del traffico HTTPS acquisito tramite il tracciamento del traffico di rete.

Questo non era ovviamente possibile nella versione 1.0 in quanto non vi era la possibilità di gestire il client dell'utente, ne sarebbe pienamente prevedibile quale browser l'utente utilizzerà.

Questo tema ha però introdotto, in fase di implementazione, l'onere di trovare soluzioni originali, apportando di conseguenza un fattore di complessità che ha comportato la necessità di introdurre nuove valutazioni tecniche a quelle già sviluppate. Se ne parlerà maggiormente nel capitolo dedicato alla fase di implementazione, ma in sintesi sono stati considerati due approcci: 1) riscrittura completa su tecnologia differente o 2) introduzione di un modulo aggiuntivo avente come obiettivo quello di intercettare il traffico (e relative chiavi TLS) prima che questo arrivi al meta-browserr. Questo secondo aspetto cambierebbe quindi il ruolo del meta-browser rendendolo quindi un "mediatore" tra il nuovo componente introdotto e l'utente finale.

4.3 Criticità residue

Seppure nella fase 2.0 siano stati risolti molti punti critici, ci sono alcuni aspetti delicati che non trovano un risoluzione immediata e che "affliggono" sia la fase 1.0 che la 2.0.

L'obiettivo del progetto di tesi è quello di creare uno strumento server-side in ottica "SaaS" (*Software as a service*) che permetta di effettuare acquisizioni forensi di contenuti Web, ed è proprio nel concetto di "server-side" che risiede il punto critico.

L'intento di fornire un servizio server-side ha come scopo di rendere fruibile agli utenti un prodotto già confezionato dotato di tutte le precauzioni forensi necessarie e che risulti di immediata fruizione senza la necessità, da parte degli operatori, di configurare degli ambienti dedicati e di seguirne il loro aggiornamento costante. Proprio questo aspetto rende evidente un punto critico: l'indirizzo di uscita.

Avere un sistema di acquisizione Web SaaS ha come conseguenza quella di creare un'infrastruttura in un ambiente strutturato, ridondato in alta affidabilità e questo quindi apre le porte a svariate soluzioni: installazione della propria infrastruttura presso un data center (di proprietà o con spazi affittati) oppure sfruttare i servizi in Cloud come per esempio Amazon Web Services o Aruba Cloud.

Nel caso che vede l'installazione presso un proprio data center si presenta il problema della facile identificazione dell'indirizzo IP di uscita, che sarà limitato al set di indirizzi acquistati e quindi facilmente riconoscibili, comportando di conseguenza il rischio di essere inseriti in "black list" e quindi tagliati fuori dai contenuti da acquisire, se non addirittura di subire vere e proprie azioni di "anti-forensics" con lo scopo di veicolare contenuti differenti per coprire i contenuti originali.

Nel secondo caso, che vede come soluzione quella di adottare servizi in cloud, si potrebbe risolvere il problema degli IP riconoscibili potendo fruire (a seconda del servizio cloud adottato) di indirizzi IP dinamici generici e sempre differenti che quindi risulterebbero non facilmente identificabili. Si pensi per esempio agli "Elastic IP" di Amazon i quali sono usati da una moltitudine di servizi online, il cui blocco risulterebbe assai delicato, in quanto bloccarli vorrebbe dire tagliare fuori buona parte della rete Internet moderna.

Ma, pur arginando egregiamente il problema dell'indirizzo IP di uscita, possono esserci ancora diversi problemi:

- limiti infrastrutturali della soluzione scelta che potrebbero non permettere la necessaria personalizzazione del sistema
- costi non sempre sotto controllo
- IP di origine non coerente con la nazione di interesse (ES: Amazon per l'Europa ha istanze in Irlanda) rendendo quindi debole la prova acquisita in quanto la fruizione del contenuto non è avvenuta dalla regione interessata e quindi al di fuori del contesto legislativo nazionale. Si pensi per esempio ad un'acquisizione di un contenuto proveniente da un sito di scommesse on-line; in Italia questi siti sono limitati dall'*art. 1, comma 50, della Legge 27 dicembre 2006, n° 296*, che ne vieta la fruizione se non legittimamente autorizzati dalla AAMS (Amministrazione autonoma dei monopoli di Stato). Quindi in base al contesto in corso, un'acquisizione effettuata al di fuori dai confini nazionali italiani risulterebbe potenzialmente nulla ai fini legali.

A fronte di tali criticità si è quindi vista la possibilità di replicare quanto già effettuato da altre soluzioni forensi, come per esempio DEFT Linux o CAINE Linux, che forniscono un completo sistema forense pronto all'uso effettuando il boot direttamente da dispositivo USB esterno o da unità ottica DVD. Questa modalità prende il nome di "sistema Live" e non prevede nessuna installazione se non l'avvio al boot di un sistema pre caricato, rendendo quindi facile ed immediata la fruizione del sistema senza la necessità per l'operatore di attuare nessuna installazione o relative configurazioni.

Questa possibilità renderebbe quindi ogni singolo operatore indipendente da qualsiasi infrastruttura, aggirando in maniera immediata il problema dell'identificazione dell'indirizzo IP di uscita, ma come la metafora della coperta troppo corta insegna, si vengono però ad introdurre nuove criticità:

- infrastruttura di rete sotto controllo dello stesso operatore e quindi soggetta a legittimi dubbi di autorevolezza
- mancanza di un garante terzo che ne possa garantire l'affidabilità di coerenza che temporale
- mancanza di un repository su servizi di terze parti in grado di mantenere lo storico e le relative garanzie senza la possibilità di fornire il riscontro di quanto esportato con quanto presente sul repository che ne dia garanzia di coerenza

- senza un repository sarà quindi necessario esportare le acquisizioni ogni qual volta che si conclude un'attività, con il relativo rischio di effettuare errori in grado di rendere nulla la prova acquisita.

4.4 Conclusioni

Webidence ha avuto un processo di sviluppo in più fasi e ognuno ha portato alla luce i propri potenziali e le proprie criticità. La fase 1.0, seppure molto lodevole dal punto di vista degli obiettivi e del codice prodotto, ha purtroppo dato come unico contributo (ma comunque importante) quello di sviscerare le criticità e i relativi limiti, riuscendo a portare nuove conoscenze utili per la fase di progettazione realizzazione della fase 2.0; quest'ultima, attraverso l'adozione di soluzioni drasticamente differenti rispetto alla fase 1.0, ha potuto seguire un percorso più sicuro e utile al risultato che si voleva raggiungere.

Webidence 2.0, pur risolvendo praticamente tutti i problemi della prima versione, non risulta però immune agli aspetti che la natura server-side introduce. Questi aspetti legati alla sua natura server-side, pur essendo punti delicati, non si possono catalogare come veri e propri "problemi", ma piuttosto elementi importanti dei quali è necessario avere la giusta consapevolezza, nella prospettiva del tipo di acquisizione che si andrà a svolgere; e ciò in quanto, seppure è vero che certi contenuti possono essere offuscati avendo conoscenza degli indirizzi IP di uscita, è anche vero che sono casi limite e facilmente identificabili per i quali all'occorrenza si possono dedicare risorse alla loro risoluzione. Inoltre, è sempre bene tenere a mente quelle che sono le casistiche di maggiore utilizzo, in quanto non è mai una buona scelta investire risorse economiche e/o tecnologiche per arginare problematiche si importanti, ma non supportate da un adeguato numero di richieste. Risulta quindi una strategia voluta la scelta di non soddisfare tali necessità e farne carico solo nel momento in cui intervengano.

Nel prossimo capitolo si andrà a presentare la fase di implementazione della soluzione 2.0 che ha fatto tesoro di quanto esposto dalle criticità in precedenza illustrate per selezionare le tecnologie che meglio potessero sposare la causa e gli obiettivi del progetto e allo stesso tempo fossero in grado di aggirare i problemi della versione precedente.

5 Webidence 2.0 - Fase realizzativa

Si illustreranno in questo capitolo le implementazioni adottate per la realizzazione della "Fase 2", trattandole con un approccio di analisi più approfondito rispetto a quanto mostrato nel capitolo precedente, così da fornire maggiore chiarezza sul lavoro svolto dal punto di vista tecnico e realizzativo.

Webidence 2.0 si differenzia in modo abbastanza marcato da quanto sviluppato nella Fase 1; nella prima versione erano state investite molte risorse nella scrittura di un codice che permettesse di raggiungere le finalità prefissate nella maniera più indipendente possibile da tecnologie di terze parti. Diversamente, nella Fase 2, si è focalizzata l'attenzione più su uno studio che ha previsto sin da subito l'integrazione tra differenti tecnologie in maniera organica tra loro, sia sviluppando in proprio quanto necessario, sia facendo uso di strumenti preesistenti su cui attuare le dovute configurazioni e/o modifiche.

Sulla struttura tecnologica predisposta si è provveduto a replicare la metafora del "meta-browser annidato" vista nel precedente progetto, non più sviluppandolo come una Web application, ma come software stand-alone da eseguire remotamente e da rendere fruibile nel browse dell'utente.

Di seguito, del presente capitolo, si tratteranno quindi le tecnologie, le configurazioni, le strategie e gli studi prodotti, riportando eventuali snippet di codice ove si necessitasse di maggior approfondimento tecnico.

5.1 Approccio

Verso la fine della Fase 1, sono state progressivamente affrontate criticità di cui si è dettagliatamente discusso e si coglierà ora l'occasione per farne una cronostoria che possa aiutare a comprendere il passaggio da "Fase 1" a "Fase 2".

Poiché i problemi non arrivano tutti in una volta, ma si manifestano gradualmente col tempo, nel tentativo di arginarli si era tentato inizialmente un approccio conservativo, nel vano tentativo di non "gettare" quanto di buono si era già avuto modo di sviluppare. Si cercava quindi di risolvere i problemi che strada facendo si affacciavano durante l'evoluzione del progetto.

Il primo grande problema con il quale ci si è dovuti confrontare fin dal "giorno zero", è stata la difficoltà di riuscire a gestire le incompatibilità introdotte dai siti stessi: incompatibilità volutamente introdotte per renderne ardua la fruizione da sistemi automatici (Es: bot) o che veicolassero i contenuti tramite "proxing indiretti" (così come era la Fase 1). Altre incompatibilità erano invece presenti per intrinseche complessità implementative del sito stesso, come per esempio script particolarmente sensibili a re-inserirsi in contesti differenti da quelli di origine.

Si è quindi sin da subito manifestato il problema di creare una strategia che fosse la più indipendente possibile dal singolo sito, in quanto tentare di studiare strategie ad hoc per ogni contenuto che manifestasse problemi di compatibilità risultava quantomai infattibile. Questo argomento ha fatto la sua invasione ripetutamente tra le tematiche di sviluppo, introducendo di volta in volta soluzioni più o meno fantasiose che ci si è visti costretti ad accantonare. Di seguito due esempi:

- al posto di fornire al client il codice HTML generato dalla semplice richiesta HTTP, si era pensato di usare PhantomJS (o anche un altro browser) eseguito lato server per veicolare all'utente i contenuti del sito una volta che il DOM fosse stato manipolato, e di volta in volta inviare al browser eseguito lato server gli eventi generati dall'utente durante la sua interazione con la pagina. Questa soluzione ha mostrato sin da subito di essere troppo debole e instabile, in quanto risultava abbastanza arduo riuscire a veicolare stream asincroni al browser dell'utente, poiché non sempre l'esecuzione di uno script poteva essere il medesimo tra i due browser e in questo determinato caso risultava impossibile fornire al browser utente il dato richiesto. Inoltre anche i tempi di esecuzione nel trasferire gli eventi dell'utente al browser in esecuzione lato server per poi girare

a sua volta le stesse richieste al server remoto per poi far ritornare i dati al client dell'utente che le aveva generate, introduceva un overhead temporale considerevole,

- usare il browser lato server per effettuare il rendering della pagina elaborata e trasmettere l'immagine ottenuta al client per poi tracciarne l'attività su di essa, per poi replicare le medesime azioni svolte dall'utente internamente al browser in esecuzione sul server. La realizzazione così concepita introduceva non poche complicazioni da gestire singolarmente che avrebbe reso la sua realizzazione alquanto intricata, poiché gli eventi generati dall'utente sarebbero dovuti essere riportati uno a uno pedissequamente sul browser simulato a livello server, comprensivi i singoli spostamenti del puntatore. Come si vedrà in seguito, la soluzione finale che si è adottato, seppure simile a quella appena menzionata, ha comunque un approccio degli eventi decisamente differente, ma cosa ancora più importante ha permesso di non "reinventare la ruota" ma di sfruttare una tecnologia già presente e matura.

Mentre si cercavano di risolvere le tematiche di compatibilità con le varie pagine che progressivamente si aggiungevano copiosamente alla lista delle "pagine problematiche", si sono presentati nuovi e differenti temi da affrontare durante l'avanzare del progetto. Queste nuove tematiche nate durante lo sviluppo, oltre ad essere nate in seno al progetto stesso, sono intervenute anche grazie al confronto intercorso con professionisti del settore forense che, con il loro prezioso contributo da "non informatici", hanno permesso di identificare i limiti della soluzione oltre a suggerire nuove idee da implementare che però sarebbero state impossibili da realizzare mantenendone invariata la struttura.

Così da un primo approccio troppo conservativo si è passati all'approccio rivoluzionario che ha visto la decisione di scartare tutto il lavoro fatto per iniziare una nuova strada decisamente più sostenibile permettendo di ottenere maggiore flessibilità.

Giunti quindi nuovamente di fronte ad una "pagina bianca" non è stato sin da subito facile individuare la strada da seguire; risultava difficile fare piazza pulita delle idee precedenti e spesso si rischiava di cadere nel vano tentativo di cercare una strada correttiva che prevedesse la riesumazione del vecchio progetto, ma per fortuna il tempo aiuta a mettere ordine alle idee e a trovare nuove soluzioni.

Di seguito si illustreranno le tecnologie principali alla base della nuova soluzione che hanno permesso di arrivare al secondo approccio denominato “Fase 2” (o “Webidence 2.0”), lasciando al capitolo successivo il compito di illustrare le parti più rigorosamente implementative.

5.1.1 Xpra (X Persistent Remote Applications)

L'approccio introdotto dalla nuova fase si è quindi manifestato con l'intento di creare una soluzione simil “desktop remoto” (così come appunta faceva in modo simile LegalEye). Inizialmente l'idea era di fare uso del protocollo VNC, ma sin da subito da tale soluzione ci si aspettavano svariati limiti: di performance, di personalizzazione, di compatibilità come l'incapacità di veicolare l'audio o la fruizione tramite client sviluppati facendo uso di “applet Java” (tecnologia ormai decisamente deprecata). Risultava però possibile aggirare l'utilizzo dell'applet Java facendo eventualmente uso del client noVNC (https://kanaka.github.io/noVNC/), un client VNC che non necessitava altro che un browser con supporto ad HTML5 e le WebSockets, ma anche questa soluzione aveva aspetti decisamente poco adatti a quanto erano le aspettative, il client in questione richiedeva comunque una server VNC in esecuzione con il relativo problema di dover gestire una sessione X11 indipendente per ogni VNC in esecuzione oltre la necessità di essere eseguito con privilegi di super user. Inoltre noVNC pur facendo uso delle WebSocket, non le gestisce nativamente e necessitava quindi di un “WebSocket proxy” separato con relativi overhead di configuraione e di carico operativo

Continuando le ricerche si quindi giunti ad un'altra soluzione: **Xpra** - multi-platform screen and application forwarding system (https://www.xpra.org/).

Figura 16: Xpra logo

Questa soluzione si è presentata fin da subito molto interessante, in quanto prevedeva caratteristiche innovative ed in linea con quelle che erano le necessità :

- sviluppato in Python (quindi facilmente personalizzabile in caso di necessità)

- client con supporto nativo ad HTML5 integrato (quindi ancora facilmente personalizzabile)
- indipendenza dalla risoluzione dello schermo, con la possibilità di gestire dinamicamente la risoluzione del virtual screen in base alle dimensioni della finestra del browser
- istanziazione di un sistema X11 di base appoggiato su di uno schermo virtuale in grado di eseguire la sola applicazione di interesse senza la necessità di eseguire un intero sistema desktop
- windows manager composito indipendente, che permette di configurare le finestre in maniera autonoma e facilmente personalizzabili tramite javascript e css
- capacità di far convivere più sessioni X11 parallelamente senza generare conflitti con altre sessioni attive
- protocollo custom "self-tuning" e relativamente "latency-insensitive" che rende il suo utilizzo adatto anche su linee non estremamente performanti in quanto in grado di generare compressioni dinamiche e separate per singole applicazioni (finestre)
- esecuzione con privilegi di base senza necessità di eseguire nulla come super user, mantenendo quindi tutto auto-contenuto in una normale sessione utente

[WKXPRA].

Altro aspetto molto interessante di Xpra rispetto ad uno normale "X forwarding", è la capacità di mantenere viva la sessione applicativa senza che questa venga distrutta in caso di interruzione della connessione, evitando quindi la perdita di dati e la relativa perdita della sessione di lavoro, potendo riprenderla trasparentemente appena ristabilita la connessione.

Le prime prove si sono eseguite con l'utilizzo di un browser classico (Firefox, Chrome, Chromium) da utilizzare remotamente annidato internamente ad un normalissimo browser moderno, riuscendo ad ottenenere ottimi risultati nella fase utilizzo, riusciendo anche a riprodurre un video su YouTube comprensivo di audio con delle perfoemance qualitativamente soddisfacenti.

Entrando nel merito delle valutazioni quantitative: per una normale navigazione su siti privi di contenuti audio/video si sono ottenute performance di consumo medo in download lato client di 180KB/sec, e uno scambio dati irrilevanti nella fase di upload.

Una navigazione più complessa che ha visto l'esecuzione di riproduzioni video e audio su youtube.com ha invece comportato un sensibile aumento dei consumi di banda, ma comunque ritenuti accettabili per la tipologia di contenuto di cui si stava fruento, riscontrando un consumo medio di banda in download nell'ordine di 650KB/sec e sempre uno scambio dati irrilevanti nella fase di upload.

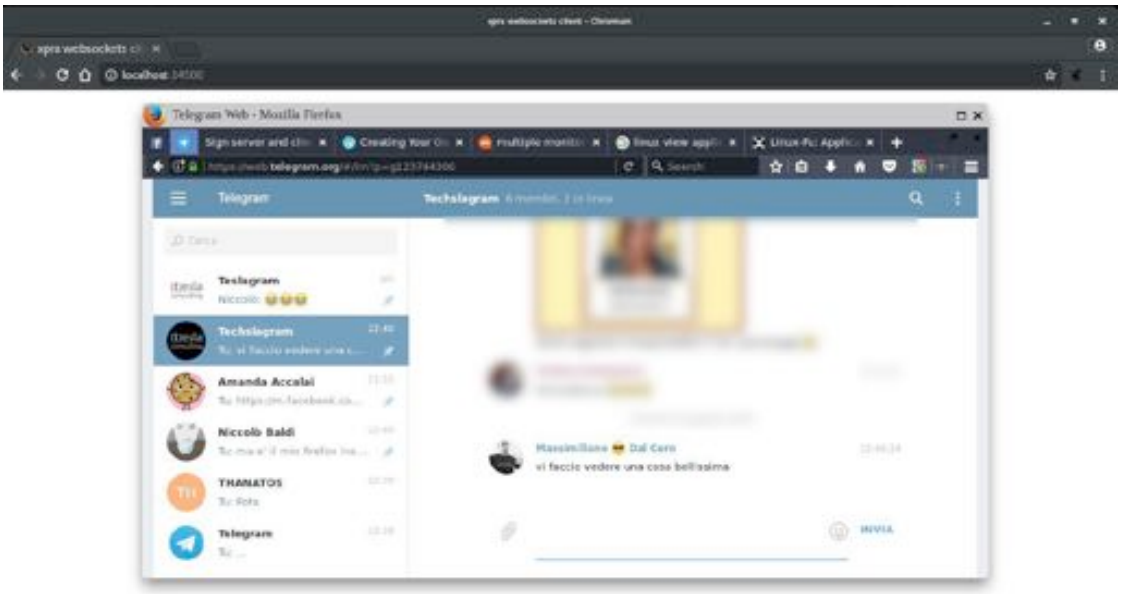


Figura 17: Xpra: Firefox annidato dentro Chromium con Telegram funzionante

La scelta finale della tecnologia da adottare per creare la sessione applicativa remota è quindi ricaduta in maniera abbastanza naturale su Xpra trovando via via conferme durante la fase di sviluppo della seconda fase. Conferme caratterizzate proprio dalla sua capacità innata di essere flessibile, leggero e sopratutto dalle elevate possibilità di personalizzazione che hanno permesso di ricreare con successo l'ambiente che si aveva come obiettivo su cui far vivere il prodotto finale. Inoltre essendo riusciti ad ottenere una simulazione con performance di rete adeguate alla fruizione del “meta-browser” , la scelta di Xpra è stata ancor più giustificata e definitiva.

5.1.2 Forensics Browser

Una volta individuato il sistema idoneo ove far vivere “l'ambiente forense”, e non potendo più contare sulla possibilità di realizzare un “meta-browser” sfruttando una

“WebApp”, si ergeva la necessità di sviluppare un vero browser forense stand-alone adatto allo scopo.

Per poter giungere allo scopo si poteva seguire più di una strada:

- configurare un browser (es: Firefox) in modalità kiosk con in più la scrittura di specifiche add-on che ne implementassero quanto richiesto
- integrare un motore di rendering in un proprio applicativo sviluppato in uno specifico linguaggio di programmazione, alcuni esempi sono:
  - CEF - Chromium Embedded Frameworks (e relativi bindings, es: cefpython) (https://bitbucket.org/chromiumembedded/cef)
  - Google Blink (https://www.chromium.org/blink)
  - Apple WebKit (https://webkit.org/)
  - Mozilla Gecko (https://developer.mozilla.org/it/docs/Gecko)
  - KHTML (https://konqueror.org/features/browser.php)
- usare un frameworks con già al suo interno un motore di rendering, esempio **NW.js** (https://nwjs.io/), **Electron** (https://electron.atom.io/), oppure il vecchio **Mozilla Prism** ormai abbandonato (http://www.mozillalabs.com/en-US/prism/); Di particolare interesse quindi sono risultati essere nw.js ed Electron, che sono due framework molto simili negli intenti, in quanto basandosi su NodeJS con l'integrazione del motore di rendering di Chromium permettono la scrittura di applicazioni desktop con solo l'utilizzo di tecnologie web oriented (html, javascript, css).

La prima possibilità si è rivelata troppo limitante, sollevando troppi dubbi in merito alla sua reale validità in linea con lo scopo del presente progetto, e non è quindi mai stata presa realmente in considerazione.

La seconda possibilità è consistita in un primo esperimento di sviluppo di una soluzione embrionale utilizzando CEFPython, implementando la possibilità di effettuare l'intercettazione delle richieste per passare ciascuna di esse alla libreria python urllib3 che permette di fare uso di PyOpenSSL, quest'ultima necessaria per effettuare le intercettazioni delle chiavi SSLT/TLS.

Poiché la necessità primaria era fornire una **prima implementazione** con l'obiettivo di determinarne i limiti e le possibilità, si è quindi optato per una soluzione ibrida arrivando a selezionare la terza opzione tra quelle poc'anzi riportate. Dovendo scegliere tra **NW.js** ed **Electron** ed essendo due soluzioni molto simili tra loro, la scelta finale è

stata effettuata in base ai reali vantaggi che una soluzione forniva rispetto all'altra, trovando (tardivamente) in NW.js la soluzione che più facilmente permetteva di creare un browser grafico altamente personalizzato oltre alla possibilità nativa di tracciare le chiavi SSL/TLS.

Sin da subito, ancora prima di scrivere una riga di codice, si è studiato l'approccio migliore da dare all'interfaccia del browser, che dovendo essere puntualmente specifico per un determinato scopo (forense) non avrebbe potuto presentare un'interfaccia generica.

L'implementazione del browser forense è stata quindi prevalentemente guidata dallo scopo di dar vita ad un prodotto che fosse il più minimale possibile, riducendone di conseguenza il margine di interazione con l'utente, con l'obiettivo di mantenerne così un procedimento lineare privo di quelle interazioni che potessero essere difficili da seguire dalla visione della sessione video. Sono quindi state eliminate le frecce di navigazione che permettono di spostarsi avanti e indietro nella history, in tal modo privando l'attività di navigazione da salti repentini difficili da seguire e obbligando l'operatore a reinserire l'url nella barra dell'indirizzo, esplicitando quindi la propria intenzionalità di cambiare pagina. Si sono inoltre eliminate le funzionalità superflue come per esempio la possibilità di creare i bookmarks, la tendina della history, il tasto destro del mouse e tante altre frivolezze di contorno che ormai ogni browser è pieno.

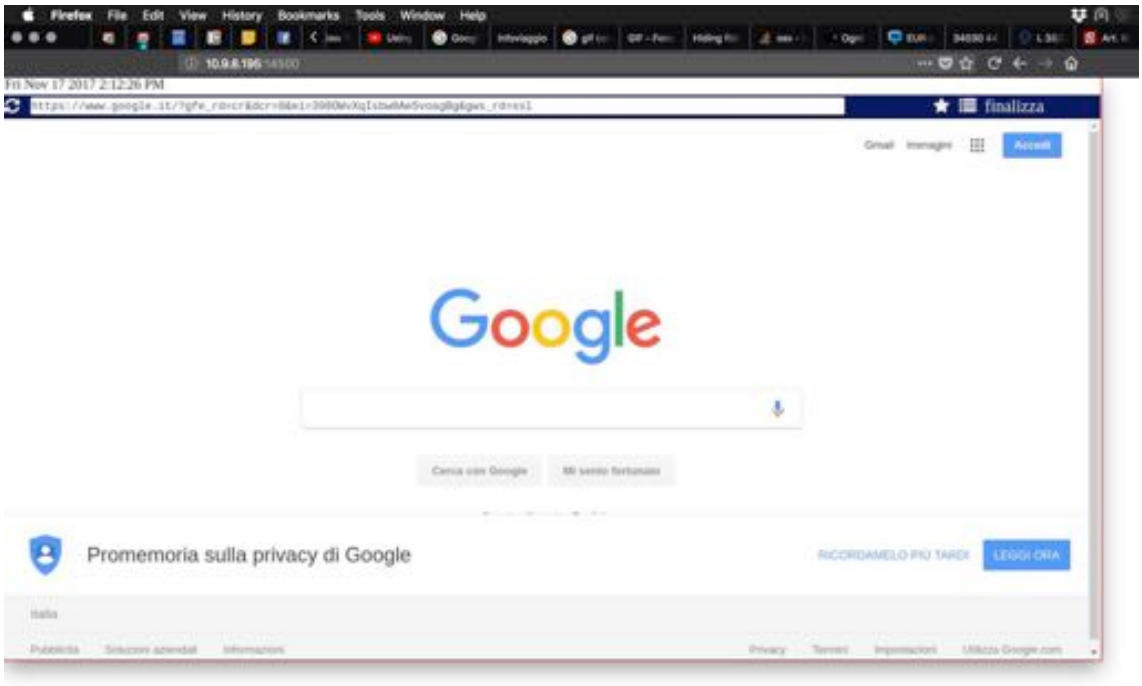
Oltre alla rimozione delle varie caratteristiche superflue, sempre nell'ottica di fornire un prodotto atto allo scopo, sono stati introdotti tre pulsanti idonei ad attivare tre specifiche funzionalità di primaria importanza:

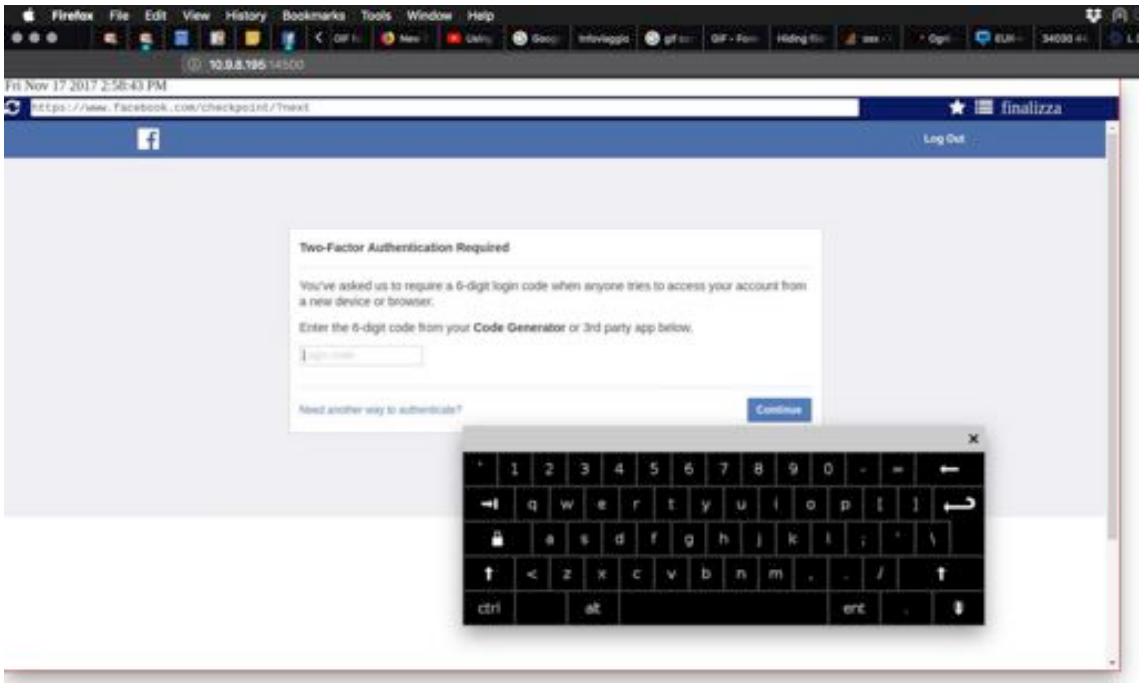
- tastiera virtuale
- selettore della lingua
- finalizzazione (acquisizione)

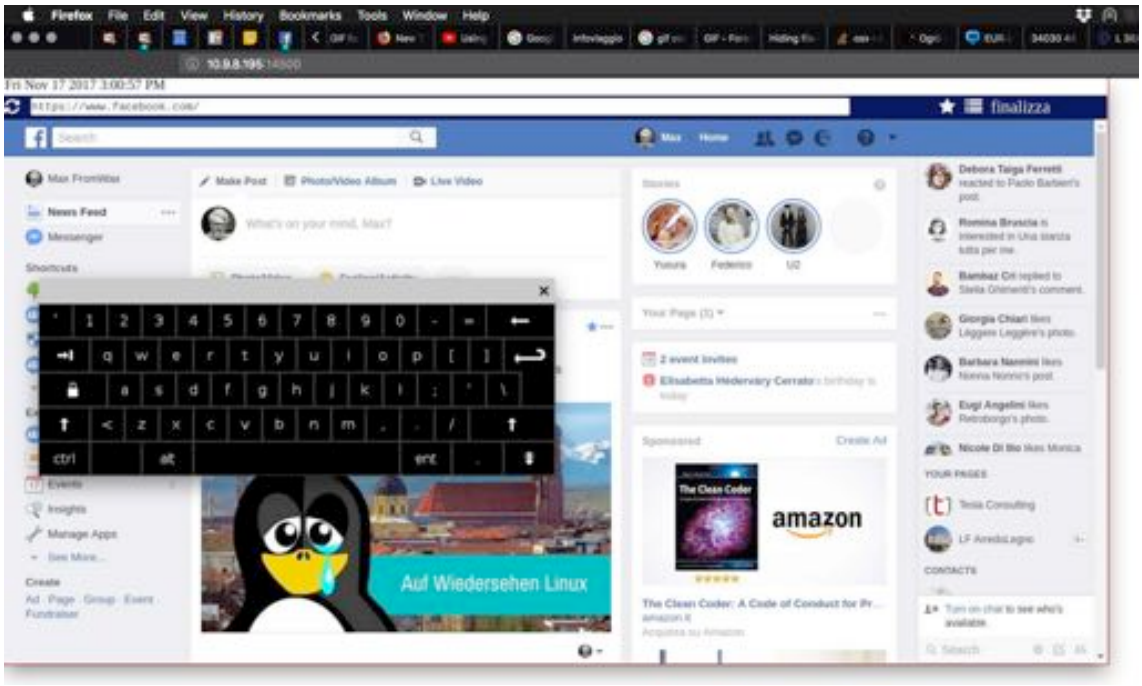
Nota: Per una prima spartana valutazione si è scelto Electron per implementare la prima versione del "forensics-browser" in quanto si era considerato un progetto supportato da una community florida e da nomi altisonanti del panorama IT, ma seppur utile allo studio della user interface e alle configurazioni da applicare alle tecnologie sottostanti atte a formare tutto il sistema e a renderlo fruibile da un normale browser web, questa prima implementazione non ha però risposto alle necessità legate alla

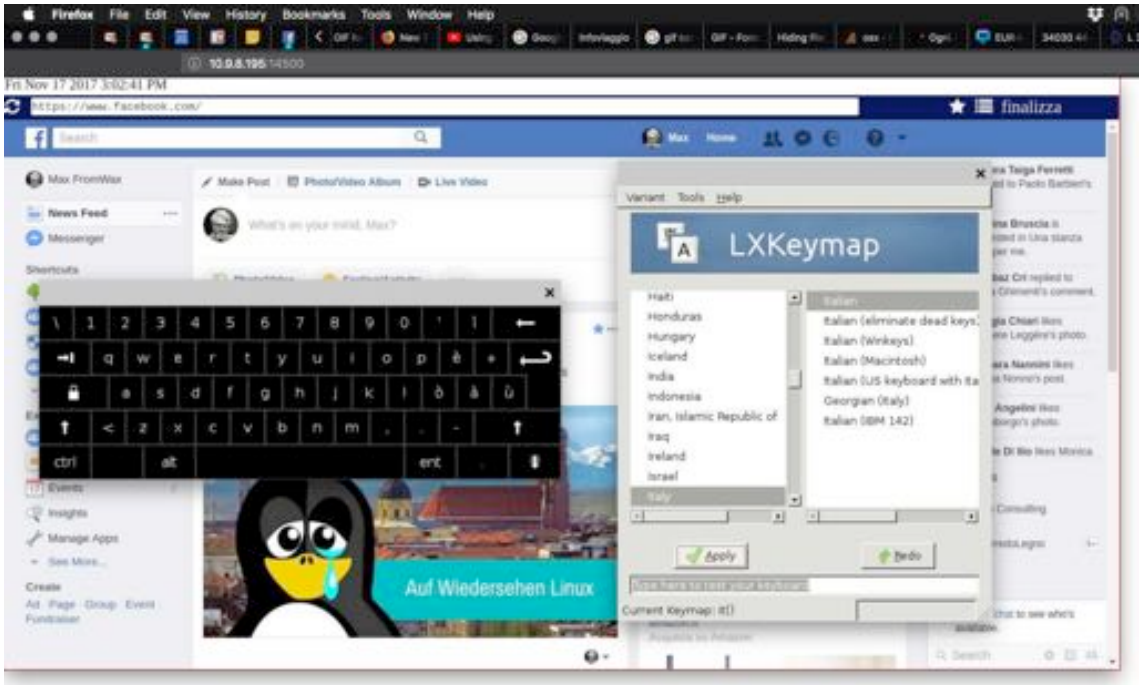
navigazione su protocolli cifrati (HTTPS). Si è infatti scoperto tardivamente che Electron non espone le dovute funzionalità utili a catturare le chiavi SSL/TLS necessarie alla decifrazione del traffico di rete acquisito, individuando solo successivamente da un test che invece le necessarie funzionalità erano presenti in NW.js (di cui non si trova traccia in nessuna documentazione).

Essendo questo un aspetto molto importante e ricco di sfumature, e che ha portato all'introduzione di nuove tecnologie e strategie, si rimanda l'esposizione dell'argomento al capitolo dedicato dove si approfitterà per discuterne più dettagliatamente.


Firefox File Edit View History Bookmarks Tools Window Help
Fri Nov 17 2017 2:12:26 PM
finalizza
Google
Promemoria sulla privacy di Google


Firefox File Edit View History Bookmarks Tools Window Help
Fri Nov 17 2017 2:58:43 PM
finalizza
Log Out
Two-Factor Authentication Required
You've asked us to require a 6-digit login code when anyone tries to access your account from a new device or browser.
Enter the 6-digit code from your Code Generator or 3rd party app below.
Continue


Auf Wiedersehen Linux
finalizza
amazon


Auf Wiedersehen Linux
finalizza
LXKeymap

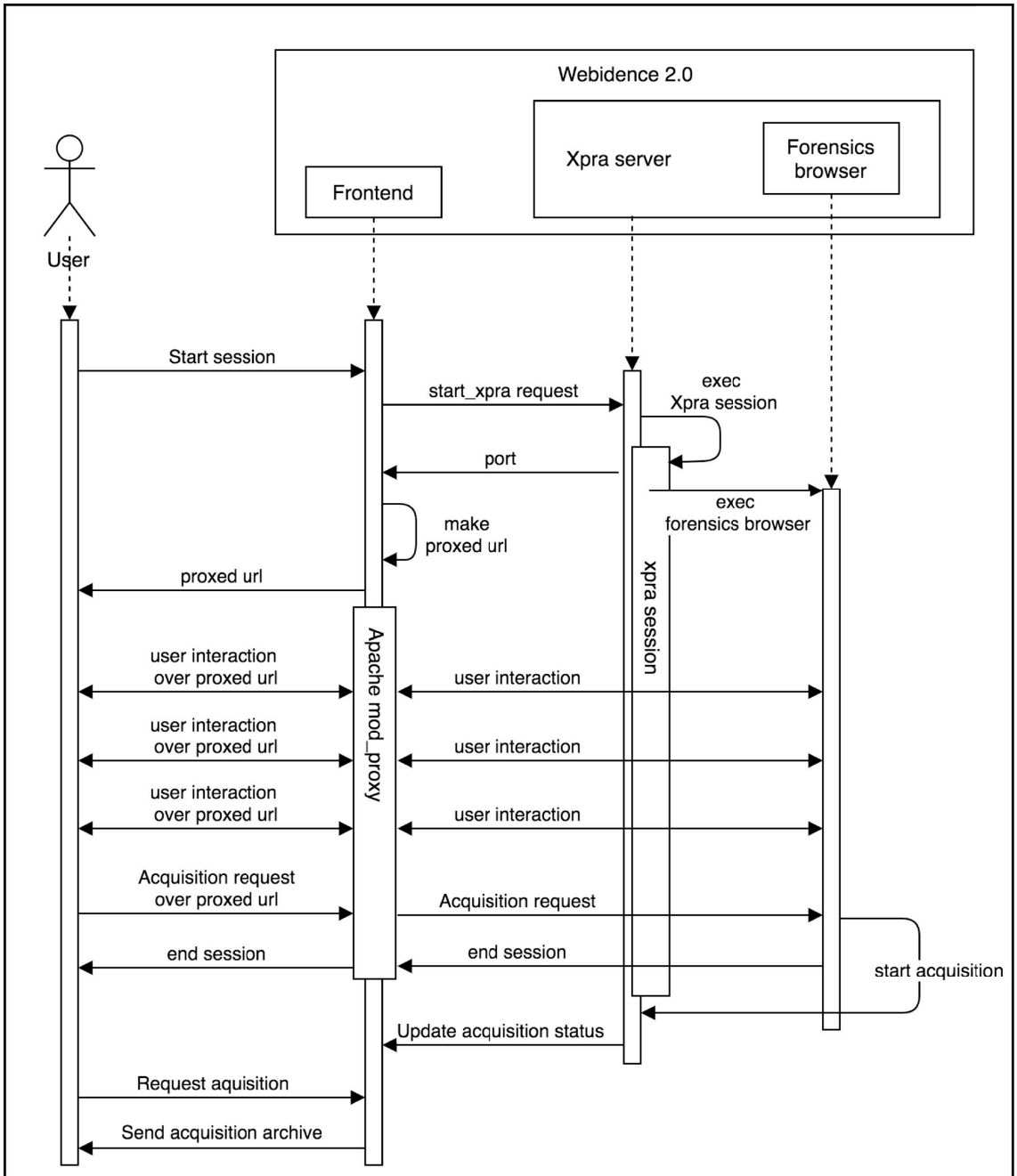


Figura 18: Webidence 2.0: Sequence Diagram generale

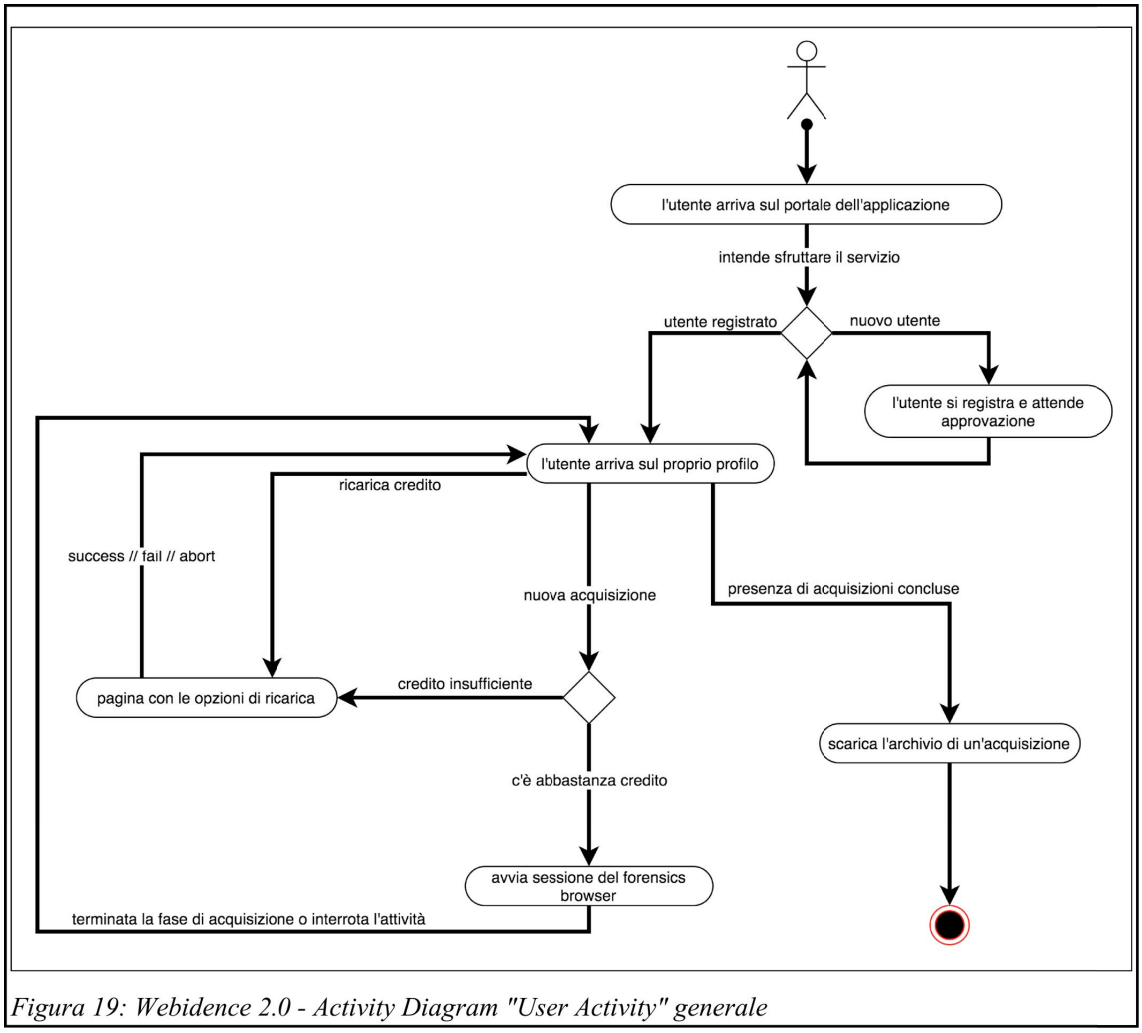


Figura 19: Webidence 2.0 - Activity Diagram "User Activity" generale

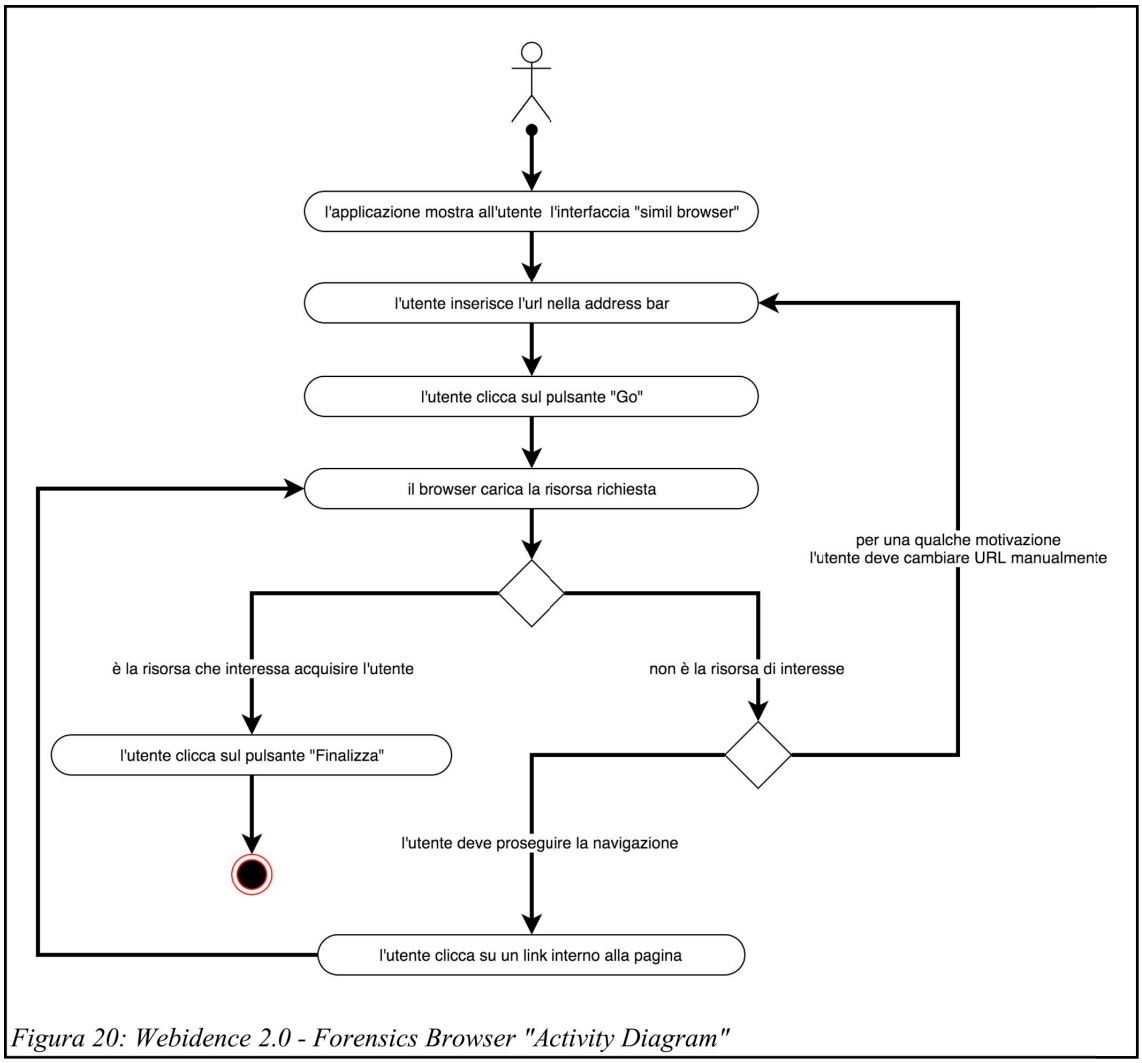


Figura 20: Webidence 2.0 - Forensics Browser "Activity Diagram"

5.2 Implementazione

Dopo aver appena illustrato l'approccio adottato oltre ad aver introdotto le due principali tecnologie adottate (Xpra e "Forensics Browser" basato su Electron), si proseguirà il presente documento con l'illustrate i dettagli implementativi adottati, cogliendo l'occasione di approfondire alcune tematiche oltre che sugli strumenti tecnici precedentemente introdotti e che sono rimaste inespresse.

Le due macro categorie sulle quali ci si concentrerà nel presente capitolo saranno il **sistema di frontend** e **il sistema di backend.** Nel sistema di frontend si illustrerà la parte relativa all'implementazione a più stretto contatto con l'utente; quindi si parlerà della parte server side dedicata alla gestione delle sessioni utente e il relativo instradamento delle stesse verso il sistema di backend così come schematizzato in Figura 18. Nella parte relativa al backend si parlerà invece delle scelte e delle implementazioni necessarie a rendere funzionante la soluzione a livello server side.

Sarà inoltre oggetto di descrizione per entrambe le due categorie la parte relativa al Forensics Browser, in quanto formalmente vive all'interno del backend come processo a stand-alone, ma la sua reale fruizione da parte dell'utente avviene nella parte di frontend. Si parlerà quindi della sua implementazione nella fase di backend e della sua fruizione ad alto livello nella parte di frontend.

5.2.1 Sistema di frontend

Nel momento della stesura del presente documento la parte relativa alle configurazioni server dedicate al frontend non è ancora stata svolta, in quanto sarà oggetto di lavori futuri. Si descriverà quindi di seguito una panoramica di massima di quanto si è progettato sulla carta.

Il sistema di frontend avrà il compito di gestire i login, i crediti, ma sopratutto veicolare al momento opportuno le richieste di acquisizione provenienti dall'esterno avviando una opportuna comunicazione verso i sistemi interni che ospitano tutta la parte di backend che sarà descritta successivamente. Per permettere questa comunicazione tra frontend e backend si possono usare svariati strumenti, ma nella fase attuale di progetto la scelta per il predetto task è ricaduta su RabbitMQ (https://www.rabbitmq.com/). RabbitMQ è un message-oriented middleware (detto anche broker di messaggistica) che implementa il protocollo Advanced Message Queuing Protocol (AMQP). RabbitMQ è una componente software in esecuzione come servizio (demone) che impersonifica il ruolo di “ufficio postale” riproducendo a tutti gli effetti il classico paradigma “producer consumer” su una coda FIFO. Questo demone dovrà quindi essere installato sul server di backend per permettere di riceve i messaggi e di instradarli al relativo processo interessato a ricevere il messaggio su una specifica coda.

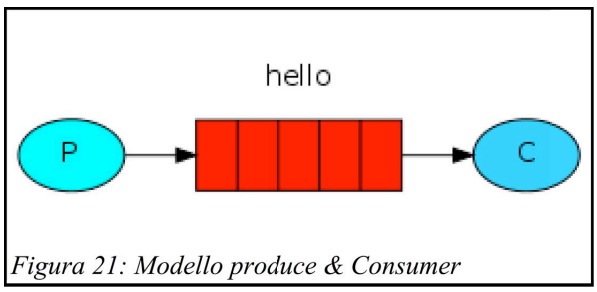


Figura 21: Modello produce & Consumer

Su questo canale di comunicazione così stabilito, si veicoleranno quindi le istruzioni da inviare al backend con i dati utili relativi alla sessione da eseguire e a sua volta fornirà le informazioni necessarie al frontend sulla porta da utilizzare per effettuare la

connessione con la sessione Xpra tramite il modulo proxy del demone HTTP in uso sul server di frontend.

Una volta che il backend avrà ricevuto le informazioni relative all'esecuzione della sessione di acquisizione, e prima di inviare la dovuta risposta al frontend, eseguirà la sessione Xpra nella quale fare eseguire il forensics-browser e i suoi relativi moduli. Da questa esecuzione potrà ottenere l'informazioni relative alla porta di comunicazione usata da Xpra per veicolare le richieste HTTP e Websocket e che sarà quindi inviata al frontend nel messaggio di ritorno.

Una volta stabilita la comunicazione bidirezionale tra frontend e backend, all'operatore sarà mostrato lo stream grafico del forensics browser e su di esso potrà operare fino a quando non attuerà la scelta di "finalizzazione" facendo uso dell'omonimo pulsante. A quel punto il backend avvierà tutte le pratiche necessarie a concludere l'acquisizione e interromperà la sessione Xpra effettuando la completa pulizia della sessione utente così da rendere tutto nuovamente fruibile per gli operatori successivi.

5.2.2 Sistema di backend

Linux

Si è implementato il sistema di backend su di un sistema operativo Linux based, la cui valutazione puntuale è stata influenzata da preferenze personali che hanno visto prediligere una distribuzione di base Debian, si è quindi sviluppata la soluzione sia in un ambiente basato su Ubuntu Desktop, sia implementando una virtual machine con sopra installata Debian Sketch priva di qualsiasi sistema grafico.

Potendo usare un sistema operativo basato su Linux si sono ottenuti svariati vantaggi (presenti e futuri):

- separazione utente dei processi, così da poter creare i singoli ambienti in maniera netta per ogni sessione di acquisizione,
- possibilità di acquisizione del traffico di rete basato su UID (User ID)
- possibilità di implementare potenzialmente in futuro ambienti isolati e protetti tramite sandboxing potendo sfruttare strumenti nativi come chroot, container LXC, CGroups o implementazioni di terze parti più strutturate come per esempio Docker

- presenza nativa delle librerie X.org necessarie per poter usare Xpra
- presenza di demoni audio (es: pulse) che permettono anche in assenza di schede audio fisiche di ridirigere il flusso audio verso un processo eseguito a livello utente che consenta la registrazione della sessione video comprensiva di suoni in maniera puntuale per ogni utente
- flessibilità di personalizzazione del sistema potendo avere nativamente a disposizione svariati linguaggi a disposizione
- sistema open-source e relativa libertà da licenze commerciali (e relativi costi)

Xpra

Come già ampiamente descritto in precedenza, come ambiente di esecuzione in cui far vivere il "forensics-browser" è stato scelto Xpra, la cui fase di configurazione nel sistema di backend ha visto come prima cosa l'installazione tramite appositi repository che ne garantissero l'allineamento all'ultima versione disponibile, in quanto essendo un progetto molto attivo vi sono continui rilasci con relativi bug fixing per cui risulta spesso importante seguire gli aggiornamenti. Essendo un progetto di "nicchia", i pacchetti disponibili tramite i canali della distribuzione non sono puntualmente allineati è quindi per questo preferibile aggiungere i repository ufficiali del progetto.

Come già precedentemente illustrato, Xpra offe un sistema persistente per l'esecuzione remota di applicazioni, sviluppato in Python e che offre svariati metodi per riuscire dialogare con la parte server tra cui un client nativo in HTML5 facente uso di Websocket.

La sua implementazione in Python è stata una delle motivazioni della scelta di Xpra, in quanto nella valutazione della tecnologia si ipotizzava la possibilità all'occorrenza di poter "aprire il cofano" con l'intento di implementare delle modifiche in modo rapido e agevole; eventualità che nei fatti non si è presentata, ma che in futuro potrebbe porsi e l'aver scelto una tecnologia basta su Python potrebbe aiutare in tal senso.

Se per la parte Python non si sono apportate modifiche, così invece non è stato per la parte di codice del client HTML5-based. Sono state apportate modifiche rese necessarie per renderlo conforme alle necessità di progetto. Di seguito si mostrerà una panoramica delle modifiche effettuate e delle relative motivazioni, ma per poter inquadrare nel modo migliore l'argomento risulterà innanzi tutto fondamentale introdurre la struttura del client HTML5 e le sue peculiarità intrinseche.

Ogni finestra applicativa in esecuzione viene rappresentata nella pagina HTML come entità a se stante e rappresentata da un tag <canvas> al cui interno viene renderizzato il contenuto binario (es: jpeg) inviato dal server al client. Questa soluzione adottata da Xpra rende particolarmente conveniente l'invio dei dati che rappresentano le singole finestre applicative, in quanto non vi sarà la necessità di inviare un'immagine complessiva e delle relative sovrimpressioni delle singole finestre. Questo genera un vantaggio nella fase di spostamento della finestra, che non genererà nuovi flussi dati in quanto essendo un componente separato la sovrimpressione delle singole finestre non ha un impatto a livello di rendering e quindi non si renderà necessario nessun "refresh". Inoltre anche l'utilizzo della singola finestra non influenza le altre e quindi il contenuto da inviare al client sarà relativo alla singola finestra interessata. Inoltre le parti dell'immagine interessate da refresh sono solo proprio quelle dove avviene l'interazione, quindi se si apre un menu contestuale l'unica area interessata sarà proprio quella dove compare il menu e la canvas sarà rigenerata solo con il frammento di immagine necessario.

Questa separazione di finestre in canvas separate permette inoltre di poter gestire le finestre con un "window-manager" residente nel browser e gestito tutto tramite script client-side e fogli di stile css, rendendo di conseguenza agevole la loro personalizzazione grazie proprio a questa netta separazione che non prevede modifiche a basso livello.

A tal proposito sono quindi state effettuate modifiche atte a generare alcuni comportamenti forzati per poter adempiere alle funzionalità richieste.

Ogni finestra prende un #id univoco composto dal nome dell'applicazione stessa, questo quindi permette di selezionare in maniera estremamente compatta le finestre di interesse modificando il codice javascript e css di interesse. Si è quindi impostato che la finestra del browser non possa essere ridimensionata, ne spostata, ne chiusa, rendendola inoltre sempre in background a qualsiasi altra finestra che si renda necessario aprire (esempio: tastiera virtuale o selezionatore della lingua).

Inoltre si sono disabilitate a livello globale le funzionalità di "ridimensionamento", di "massimizzazione" e di "minimizzazione" delle finestre in quanto, per decisione di progetto, ogni finestra dovrebbe essere solo spostata ed eventualmente chiusa (ad esclusione della finestra del browser che dovrà rimanere massimizzata e fissata in background); quindi per una questione di scelta implementativa dell'interfaccia utente si sono disabilitate le funzionalità sopra descritte per limitare al massimo potenziali

problemi di sovrimpressioni non volute o di perdita di controllo della finestra in caso di minimizzazione accidentale.

Forensics Browser

Il forensics-browser è il vero componente principale e cuore della soluzione. Le altre tecnologie in gioco sono si importanti, ma funzionali a rendere fruibile agli utenti proprio questo componente che risulta quindi essere il primario punto di contatto concretamente visibile da parte dell'utente.

Il forensics-browser risulta quindi il componente che più di altri permette di rendere possibile l'acquisizione forense comprendendo dentro di se, sia la parte di interfaccia utente che permetta agli utenti di compiere il compito nel modo migliore possibile, sia la vera parte di interazione con tutti i componenti della pagina di interesse:

- navigazione tramite gli hyperlink
- sorgente raw
- elaborazione del document object model (DOM)
- risorse statiche
- interazione con il sistema per attivare le funzionalità di:
  - virtual keyboard
  - selezione della lingua
  - finalizzazione dell'acquisizione

Essendo quindi un componente così centrale e delicato, è stato fortemente oggetto di studio per poter permettergli di compiere al meglio i suoi doveri e in quanto oggetto di studio principale non gli sono state risparmiate critiche a livello di funzionalità mancanti o incomplete. Saranno quindi riportate nel seguito del capitolo le criticità più interessanti riscontrate ed eventualmente quella ancora aperte e lasciate come implementazioni future.

Sono state vagliate varie soluzioni per poter creare il forensics-browser, tra cui:

- Electron (http://electron.atom.io/)
- CEFPython (https://github.com/cztomczak/cefpython)
- NW.js

Electron

Electron è la tecnologia su cui è sviluppato l'attuale versione del forensics-browser, è un framework basato su NodeJS e Chromium e permette quindi di offrire delle funzionalità di un browser completo più la flessibilità di sviluppo data da NodeJS, che permette di creare agilmente un'applicazione completa facendo uso solamente delle classiche tecnologie web (JS, HTML e CSS).

La soluzione attualmente sviluppata del forensics-browser basata su Electron ha pienamente rispettato le necessità richieste:

- compatibilità con gli standard moderni del web
- veloce prototipazione dell'interfaccia utente
- facilità di integrazione con la WebView con relativa possibilità di intercettare tutte le risorse caricate, codice sorgente raw e codice sorgente post elaborazione del DOM

```
> main.js:
// tracciamento di tutte le risorse caricate via http e https
app.on('ready', function() {

        var protocol = require("protocol");
        protocol.interceptProtocol('http', function(request, callback) {
                console.log({ 'url' : request.url });
        });
        protocol.interceptProtocol('https', function(request, callback) {
                console.log({ 'url' : request.url });
        });
});




> preload.js:
// predisposizione degli eventi che saranno usati
// per richiedere codice RAW e DOMContent
var ipcRenderer = require('electron').ipcRenderer;

var raw_html = "";
document.addEventListener("DOMContentLoaded", function () {
        raw_html = document.documentElement.outerHTML;
});

ipcRenderer.on('rawhtml', function(){
        ipcRenderer.sendToHost('html-raw-content' , raw_html);
});

ipcRenderer.on('request', function(){
        ipcRenderer.sendToHost('html-content' ,\
                                document.documentElement.outerHTML );
});
```

```
> render.js:
// attuazione degli eventi precedentemente impostati:
// "request" e "rawhtml"


function finalizeclick() {
        view.send("request");
        view.send("rawhtml");
}

// impostazione del listener che riceverà gli eventi precedentemente
// scatenati
view.addEventListener("ipc-message", function (e) {
        if (e.channel === "html-content") {
                var html_contents = e.args[0];
                var fs = require('fs');
                try { fs.writeFileSync('/tmp/raw_source', \
                                        html_contents, 'utf-8'); }
                catch(e) { alert('Failed to save the file !'); }
        } else if (e.channel === "html-raw-content") {
                var html_contents = e.args[0];
                var fs = require('fs');
                try { fs.writeFileSync('/tmp/dom_content.html', \
                                        html_contents, 'utf-8'); }
                catch(e) { alert('Failed to save the file !'); }
        }
});
```

Pur permettendo tutte le primarie necessità, si è purtroppo constatato solamente in una fase successiva che non esistono possibilità per integrare in un'applicazione sviluppata con Electron degli handler in grado di intercettare le sessioni SSL/TLS per estrapolarne le chiavi necessarie per la successiva decifrazione del traffico.

A tal proposito, essendo Electron basato sul framework NodeJS e nella speranza che fosse possibile veicolare le richieste HTTPS tramite degli handler alternativi anziché fare uso delle richieste eseguite in seno alla componente interna basata su Chromium, si è realizzato un codice di esempio in grado di usare le librerie native di NodeJS per intercettare la sessione SSL tramite il modulo nativo "https" per estrapolarne la client session ID e la master Key:

```
var https = require('https');
var fs = require('fs');

function parseSession(buf) {
  return {
    sessionId: buf.slice(17, 17+32).toString('hex'),
    masterKey: buf.slice(51, 51+48).toString('hex')
  };
}
```

```
callback = function(response) {
  var str = ''
  response.on('data', function (chunk) {
    str += chunk;
  });

  response.on('end', function () {
    console.log(str);
  });
}

var options = {
  host: 'github.com',
  port: 443,
  path: '/',
  method: 'GET'
};

var R2 = https.request(options, callback)
.once('socket', (s) => {
  s.once('secureConnect', () => {
    let session = parseSession(s.getSession());
    // sessionId e masterKey devono essere hex strings
    fs.appendFileSync('sslkeylog.log', 'RSA Session-ID:$ \
        {session.sessionId} Master-Key:${session.masterKey}\n');
  });
});

R2.end()
```

CEFPython

Una volta individuato che Electron non permetteva l'intercettazione della sessione SSL/TLS si sono quindi cercate nuove strade, battezzando come strada successiva una via del tutto differente rispetto ad un framework ad alto livello, ricercando invece una soluzione che potesse garantire più controllo sui singoli componenti. Si è quindi seguita la strada di CEF (Chromium Embedded Framework) e più precisamente il suo binding Python: CEFPython.

Anche in questo caso le richieste alle risorse eseguite da cefpython venivano tutte svolte dalla componente interna basata su Chromium, ma come da aspettative si è trovato il metodo di effettuare l'override degli handler permettendo di usare delle proprie classi custom.

Questo primo passo pur essendo già un punto importante, non risultava sufficiente a intercettare le sessioni SSL/TLS, in quanto le richieste https effettuate dalle librerie native di Python non permettono di estrapolare le chiavi necessarie, ne la sessione raw, si è quindi cercato una libreria alternativa che permettesse più ampi spazi di manovra,

trovando la giusta soluzione nel connubio UrlLib3 (https://urllib3.readthedocs.io) più PyOpenSSL (https://github.com/pyca/pyopenssl).

Urllib3 è una libreria Python che permette di effettuare richieste HTTP(s) fornita di comodi metodi per effettuare anche complesse richieste HTTP e poterne estrapolare da esse le informazioni necessarie in modo agevole ed elegante.

Urllib3 permette inoltre di forzare l'utilizzo della libreria PyOpenSSL invece delle librerie SSL native di Python e proprio questa caratteristica ha permesso di effettuare le dovute modifiche al codice della componente pyopenssl di urllib3 per riuscre ad estrapolare le chiavi SSL/TLS richieste.

Il file modicato è risultato essere "*/urllib3/contrib/pyopenssl.py*" .

Di seguito si riportano in grasseto le poche ma fondamentali modifiche apportate al metodo *wrap_socket* della classe *PyOpenSSLContext*:

```
def wrap_socket(self,
                sock,
                server_side=False,
                do_handshake_on_connect=True,
                suppress_ragged_eofs=True,
                server_hostname=None):

        cnx = OpenSSL.SSL.Connection(self._ctx, sock)
        if isinstance(server_hostname, six.text_type):
            server_hostname = server_hostname.encode('utf-8')
        if server_hostname is not None:
            cnx.set_tlsext_host_name(server_hostname)

        cnx.set_connect_state()

        while True:
            try:
                cnx.do_handshake()
            except OpenSSL.SSL.WantReadError:
                rd = util.wait_for_read(sock, sock.gettimeout())
                if not rd:
                    raise timeout('select timed out')
                continue
            except OpenSSL.SSL.Error as e:
                raise ssl.SSLError('bad handshake: %r' % e)
            break

        with open("/Users/max/xkeylog.log", "a") as myfile:
            myfile.write("CLIENT_RANDOM %s %s\n" %
                         (cnx.client_random().encode("hex"),
                         cnx.master_key().encode("hex"))
             )
        return WrappedSocket(cnx, sock)
```

Di seguito un esempio di codice basato su urllib3 che permette di forzare la libreria a fare uso di PyOpenSSL e che grazie alle modifiche poco sopra riportate salverà le chiavi nel file "/Users/max/xkeylog.log":

```
import OpenSSL
import urllib3
import urllib3.contrib.pyopenssl

urllib3.contrib.pyopenssl.inject_into_urllib3()
http = urllib3.PoolManager()
r = http.request('GET','https://github.com')
```

Invece di seguito si riporta un semplice esempio che fa uso della sola libreria PyOpenSSL e che non richiede nessuna modifica:

```
from OpenSSL import SSL
import sys, os, select, socket

ctx = SSL.Context(SSL.SSLv23_METHOD)
sock = SSL.Connection(ctx, socket.socket(socket.AF_INET,
socket.SOCK_STREAM))
sock.connect(("github.com", 443))
sock.do_handshake()
sock.send("""GET / HTTP/1.0\rHost: github.com\r\n\r\n""")

print( sock.client_random().hex() )
print( sock.master_key().hex() )
```

Successivamente ai test appena riportati si è quindi proceduto a creare un primo codice di esempio che riuscisse ad usare CEFPython in congiunzione delle librerie urllib3 modificate con lo scopo di salvare le chiavi SSL/TLS.

L'esempio realizzato non fa "banalmente" altro che aprire una piatta finestra senza interfaccia, con al suo interno caricata la pagina riferente all'url impostato staticamente nel codice. Seppur minimale la risorsa caricata risulta essere pienamente funzionante e navigabile come in una normale pagina caricata in un browser classico.

Di seguito il codice realizzato e a seguire gli screenshot della parte grafica:

```
from cefpython3 import cefpython as cef
import sys
import OpenSSL
import urllib3
import urllib3.contrib.pyopenssl
import re
import urlparse
from threading import Thread

urllib3.contrib.pyopenssl.inject_into_urllib3()
urllib3.disable_warnings()
http = urllib3.PoolManager()

BROWSER_DEFAULT_ENCODING = "UTF-8"


class WebRequestClient:

    _resourceHandler = None
    _data = ""
    _dataLength = -1
    _response = None

    def OnUploadProgress(self, web_request, current, total):
        pass

    def OnDownloadProgress(self, web_request, current, total):
        pass

    def OnDownloadData(self, web_request, data):
        self._data += data

    def OnRequestComplete(self, web_request):
        statusText = "Unknown"
        if web_request.GetRequestStatus() in cef.WebRequest.Status:
            statusText = cef.WebRequest.Status[webRequest.GetRequestStatus()]
        self._response = web_request.GetResponse()
        self._data = self._resourceHandler._clientHandler._OnResourceResponse(
                self._resourceHandler._browser,
                self._resourceHandler._frame,
                web_request.GetRequest(),
                web_request.GetRequestStatus(),
                web_request.GetRequestError(),
                web_request.GetResponse(),
                self._data)
        self._dataLength = len(self._data)
        self._resourceHandler._responseHeadersReadyCallback.Continue()




class ClientHandler:

    def GetResourceHandler(self, browser, frame, request):
        resHandler = ResourceHandler()
        resHandler._clientHandler = self
        resHandler._browser = browser
        resHandler._frame = frame
        resHandler._request = request
        #resHandler._cm = self.cm
        self._AddStrongReference(resHandler)
        return resHandler

    def _OnResourceResponse(self, browser, frame, request, requestStatus,
            requestError, response, data):
        return data

    _resourceHandlers = {}
    _resourceHandlerMaxId = 0
```

```
    def _AddStrongReference(self, resHandler):
        self._resourceHandlerMaxId += 1
        resHandler._resourceHandlerId = self._resourceHandlerMaxId
        self._resourceHandlers[resHandler._resourceHandlerId] = resHandler

    def _ReleaseStrongReference(self, resHandler):
        if resHandler._resourceHandlerId in self._resourceHandlers:
            del self._resourceHandlers[resHandler._resourceHandlerId]
        else:
            pass


class ResourceHandler:
    _responseHeadersReadyCallback = None
    _offsetRead = 0
    _data = None
    _heads = None
    _datalen = -1

    def go(self, ref, heads, request,callback):
        send_data = request.GetPostData()
        if request.GetMethod() == "GET" or request.GetMethod() == "HEAD" or \
                                                request.GetMethod() == "DELETE":
            r = http.request_encode_url(request.GetMethod(),
                                        request.GetUrl(),
                                        fields=send_data,
                                        headers=heads,
                                         redirect=False,
                                        preload_content=False
                                    )
        elif (request.GetMethod() == "POST" and heads["Content-Type"] == \
              "application/x-www-form-urlencoded") or request.GetMethod() == "PUT" \
                                                or  request.GetMethod() == "PATCH":
            r = http.request_encode_body(request.GetMethod(),
                      request.GetUrl(),
                      fields=send_data,
                      headers=heads,
                      redirect=False,
                      encode_multipart=False,
                      preload_content=False
                )
        elif request.GetMethod() == "POST" and "multipart" in heads["Content-Type"]:
            r = http.request(request.GetMethod(),
                      request.GetUrl(),
                      fields=send_data,
                      headers=heads,
                      redirect=False,
                      preload_content=False
                )
        else:
            r = http.request_encode_url(request.GetMethod(),
                      request.GetUrl(),fields=send_data, headers=heads,
                      redirect=False,preload_content=False
                )
        self._response = r
        self._datalen = len(r.data)
        self._data = r.data
        self._datastream = r.stream
        request.SetFlags(cef.Request.Flags["AllowCachedCredentials"] | \
                                                cef.Request.Flags["AllowCookies"])
        self._responseHeadersReadyCallback = callback
        self._responseHeadersReadyCallback.Continue()
```

```
    def ProcessRequest(self, request, callback):
        heads = request.GetHeaderMap()
        request.SetFlags(cef.Request.Flags["AllowCachedCredentials"] | \
                                                cef.Request.Flags["AllowCookies"])

        if request.GetUrl().startswith("chrome-extension"):
            self._requestz = request
            self._responseHeadersReadyCallback = callback
            self._webRequestClient = WebRequestClient()
            self._webRequestClient._resourceHandler = self
            self._webRequest = cef.WebRequest.Create(request, self._webRequestClient)
            return True
        elif not request.GetUrl().startswith("http"):
                print("URL MALFORMED: " + request.GetUrl())
                return False
        else:
            self._responseHeadersReadyCallback = callback
            self._requestz = request
            self._heads = heads
            t = Thread(target=self.go,args=(self,heads,request,callback))
            t.start()
            return True

    def GetResponseHeaders(self, response, responseLengthOut, redirectUrlOut):

        if self._requestz.GetUrl().startswith("chrome-extension"):
            assert self._webRequestClient._response, "Response object empty"
            wrcResponse = self._webRequestClient._response
            response.SetStatus(wrcResponse.GetStatus())
            response.SetStatusText(wrcResponse.GetStatusText())
            response.SetMimeType(wrcResponse.GetMimeType())
            if wrcResponse.GetHeaderMultimap():
                response.SetHeaderMultimap(wrcResponse.GetHeaderMultimap())
            responseLengthOut[0] = self._webRequestClient._dataLength
            return
        else:
            wrcResponse = self._response
            location=None
            try:
                if "location" in wrcResponse.headers:
                    location = wrcResponse.headers["location"]
                elif "Location" in wrcResponse.headers["Location"]:
                    location = wrcResponse.headers["Location"]
                if location is not None:
                    if location == "." or location == "":
                        location = self._requestz.GetUrl()
                    else:
                        locres = urlparse.urlparse(location)
                        urlres = urlparse.urlparse(self._requestz.GetUrl())

                        if locres.netloc == "":
                            if locres.path[0] == "/":
                                location = urlres.scheme + "://" + urlres.netloc + \
                                                                        location
                            else:
                                sep = "/"
                                if location[0] == "/" or self._requestz.GetUrl()[-1]\
                                                                        == "/":
                                    sep = ""

                                location = self._requestz.GetUrl() + sep + location
            except:
                pass

            if location is not None:
                redirectUrlOut[0] = location
            response.SetStatus(wrcResponse.status)
            response.SetStatusText(wrcResponse.reason)
            ct = ct0 = "none"
            if wrcResponse.headers.get("Content-Type") is not None:
                ct0 = ct = wrcResponse.headers.get("Content-Type").split(";")[0]
                response.SetMimeType(ct)
            response.SetHeaderMultimap(wrcResponse.headers.items())
            responseLengthOut[0] = self._datalen #len(self._data)
```

```
    def ReadResponse(self, data_out, bytes_to_read, bytes_read_out, callback):

        if self._requestz.GetUrl().startswith("chrome-extension"):
            if self._offsetRead < self._webRequestClient._dataLength:
                dataChunk = self._webRequestClient._data[self._offsetRead: \
                            (self._offsetRead + bytes_to_read)]
                self._offsetRead += len(dataChunk)
                data_out[0] = dataChunk
                bytes_read_out[0] = len(dataChunk)
                return True
            self._clientHandler._ReleaseStrongReference(self)
            return False
        else:
            if self._datalen < 0:
                dataChunk = self._response.read(bytes_to_read)
                if len(dataChunk) > 0:
                    bytes_read_out[0] = len(dataChunk)
                    data_out[0] = dataChunk
                    self._offsetRead += len(dataChunk)
                    return True
                else:
                    bytes_read_out[0] = len(dataChunk)
                    data_out[0] = dataChunk
                    self._response.release_conn()
                    self._clientHandler._ReleaseStrongReference(self)
                    return False
            elif self._offsetRead < self._datalen:
                dataChunk = self._data[self._offsetRead:(self._offsetRead + \
                                                         bytes_to_read)]
                self._offsetRead += len(dataChunk)
                data_out[0] = dataChunk
                bytes_read_out[0] = len(dataChunk)
                return True

            self._clientHandler._ReleaseStrongReference(self)
            return False


    def CanGetCookie(self, cookie):
        return True

    def CanSetCookie(self, cookie):
        return True

    def Cancel(self):
        pass



def main():
    sys.excepthook = cef.ExceptHook
    conf = {
        "context_menu": {
            "enabled":False
        },
        #"cache_path": "/Users/max/cache/",
        #"persist_session_cookies": True
    }
    cef.Initialize(settings=conf)
    url="https://www.facebook.com"
    browser = cef.CreateBrowserSync(url=url, window_title="UrlLib3")
    clientHandler = ClientHandler()
    browser.SetClientHandler(clientHandler)
    cef.MessageLoop()
    cef.Shutdown()

if __name__ == '__main__':
    main()
```

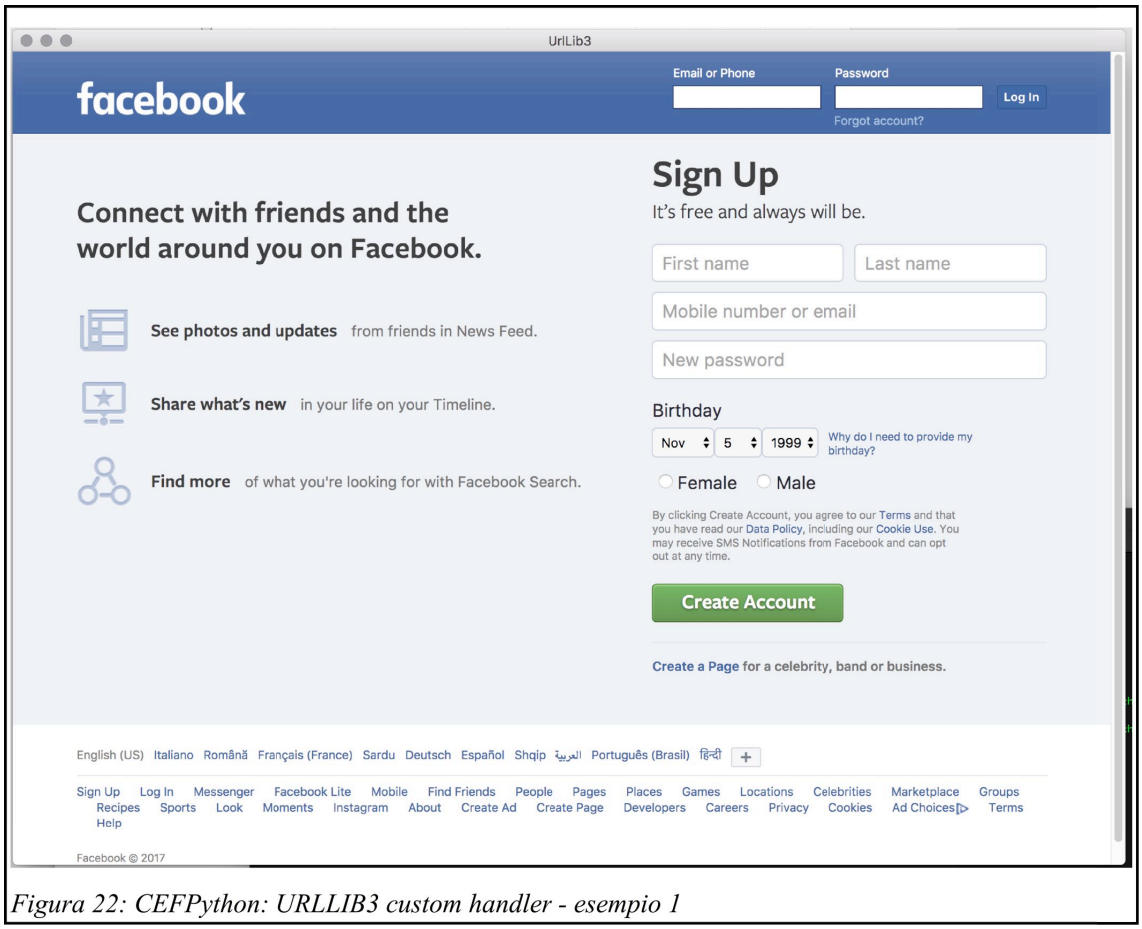


Figura 22: CEFPython: URLLIB3 custom handler - esempio 1

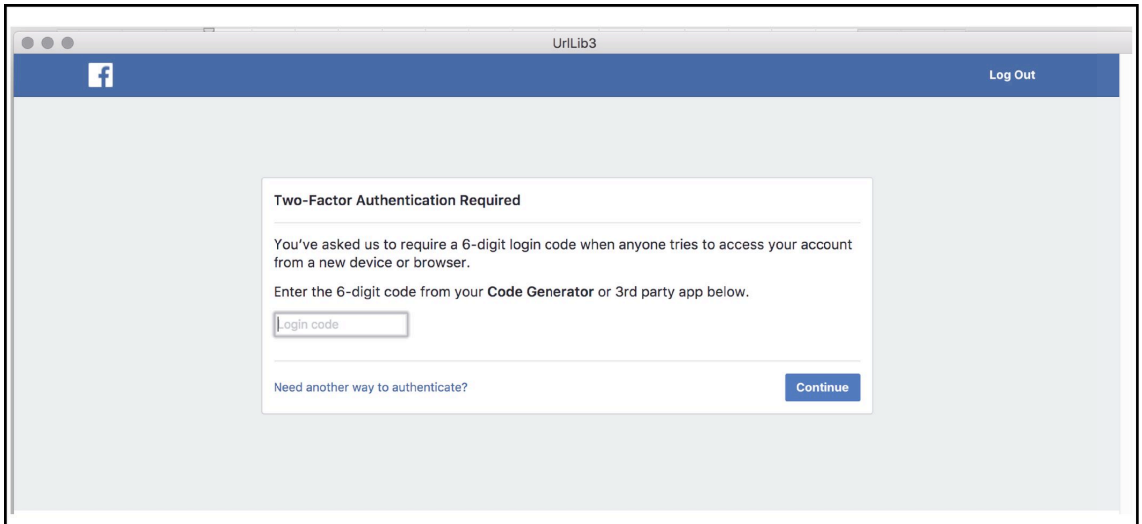


Figura 23: CEFPython: URLLIB3 custom handler - esempio 2

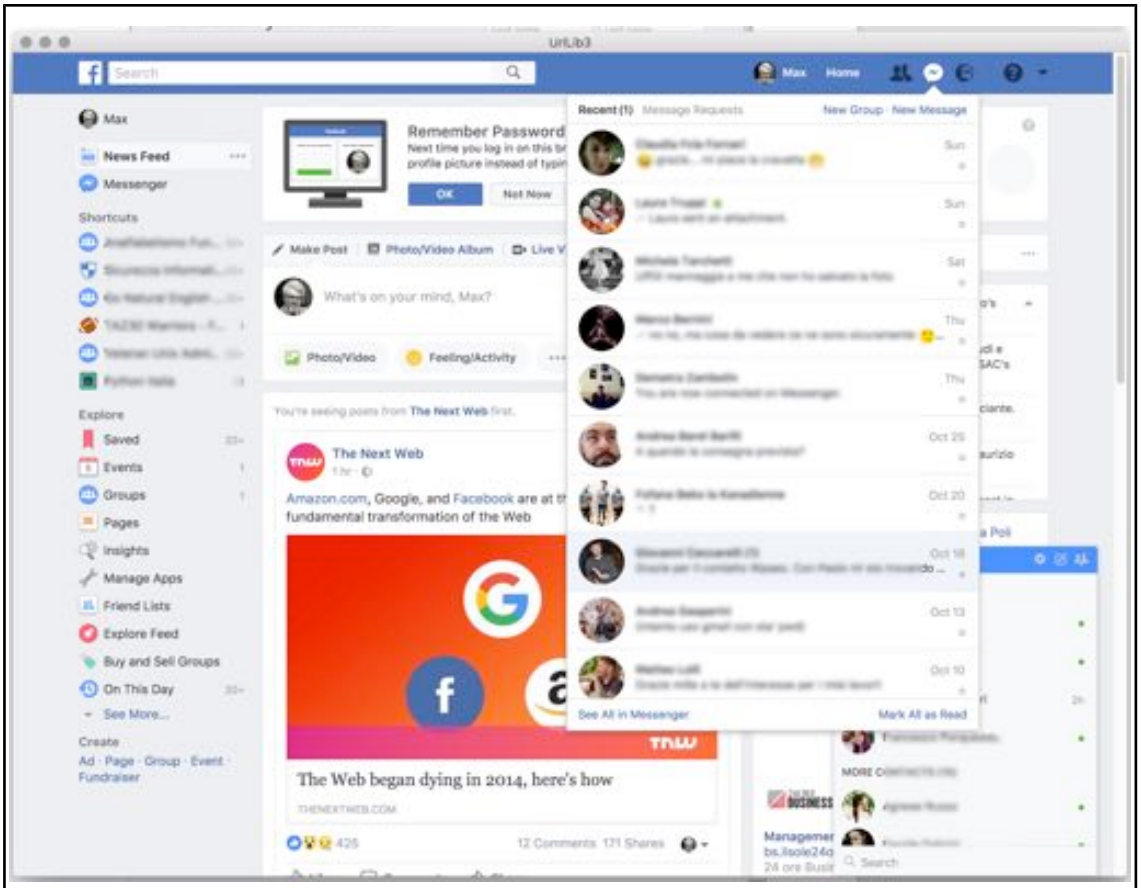


Figura 24: CEFPython: URLLIB3 custom handler - esempio 3

NW.js

Dopo tanto lavoro stimolante che ha permesso di approfondire alcuni interessanti temi in merito alle librerie e ai framework utilizzati, si è proceduto a fare un ulteriore test per non lasciare nulla di intentato. Purtroppo (o per fortuna) proprio questo ultimo test svolto come ennesimo scrupoloso tentativo di risolvere nella maniera più elegante possibile il tema delle chiavi SSL/TLS, è risultato essere l'evento capace di stravolgere il lavoro pre-esistente rendendolo potenzialmente nullo e di seguito si spiegherà dettagliatamente l'accaduto e i relativi esiti.

NW.js sia per intenti che per struttura è un framework molto simile ad Electron, entrambi sono basati su NodeJS e Chromium, anche se con le dovute differenze a livello di integrazione: NW.js integra una versione completa di Chromium, invece Electron fa uso di libchromiumcontent (https://github.com/electron/libchromiumcontent) che ne isola solo le componenti strettamente necessarie. [ED2017]

Inoltre, da una prima analisi dei due progetti, si è banalmente lasciati infliuenzare dal "curriculum" di NW.js che mostrava una minore intensità di sviluppo su github, meno clamori mediatici e risultava essere meno supportato da nomi importanti (Microsoft, Facebook, Slack, ..) come invece risultava esserlo Electron [EL2017].

Visti i limiti di Electron e vista la vicina somiglianza tra i due frameworks si era erroneamente supposto che i limiti di Electron legati alla cattura delle chiavi SSL/TLS potessero affliggere anche NW.js, ma si è invece riscontrato come il differente approccio di integrazione della componente Chromium nei due framwork avesse un'importanza fondamentale nelle dinamiche del presente progetto.

Ma per meglio capire il test svolto e il suo relativo contesto, è importante aprire una veloce parentesi su alcune caratteristiche non diffusamente note presenti in due dei browser ad oggi più noti: Chrome/Chromium e Firefox.

Questi browser permettono di esportare su file le chiavi SSL/TLS per tutte le connessioni https incontrate semplicemente impostando una variabile d'ambiente nel contesto dell'esecuzione o tramite un "command switch" passato da esecuzione tramite linea di comando.

Entrambi i browser permettono di esportare le chiavi se nelle variabili di ambiente trovano impostata la variabile denominata SSLKEYLOGFILE valorizzata con una stringa testuale contente il percorso su disco indicante dove salvare il file di log delle chiavi SSL/TLS

Chrome e Chromium invece offrono anche la possibilità di sfruttare un'apposita opzione da passare tramite linea di comando denominata "ssl-key-log-file":

```
--ssl-key-log-file=/path/del/keyfile.log
```

Proprio in merito a queste possibilità di esportazione, sia in CEF che in Electron si è riscontrato, indagando nei rispettivi codici sorgente, che tutti i riferimenti a queste due opzioni erano stati eliminati e di conseguenza non era possibile farne un uso diretto ma si richiedeva una delicata opera di revisione dei sorgenti e relativo rebuild.

Per quanto riguarda invece NW.js, nella documentazione ufficiale relativa ai "command switch" utilizzabili da linea di comando, non vi è nessuna menzione alla possibilità di fare uso dell'opzione ssl-key-log-file, ma leggendo con più attenzione quanto riportato alle pagine:

- https://github.com/nwjs/nw.js/wiki/Command-line-switches
- http://docs.nwjs.io/en/latest/References/Command%20Line%20Options/#other-chromium-options

è saltata all'occhio la seguente nota indicata nell'ultimo paragrafo:

> **And more: all the switches supported by Content API of Chromium**
>
> All the command line arguments can be predefined in the application's manifest file. See `chromium-args` in Manifest-format

Proprio la nota a pie' di pagina sopracitata ha insediato nel sottoscritto il ragionevole dubbio relativo alla concreta possibilità di riuscire sfruttare il command switch ssl-key-log-file; si è quindi proceduto a fare un primo e banale test che prevedeva la sola apertura di una semplice risorsa https.

Con inizialmente grande scetticismo e successivamente grande stupore si è potuto riscontrare che questa funzionalità risulta essere supportata e completamente funzionante rendendo di conseguenza inutile apportare modifiche invasive come la modifica del codice sorgente del framework o effettuare override degli handle.

Questo aspetto, seppure emerso tardivamente, ha riabilitato NW.js per essere il framework di riferimento per lo sviluppo del forensics-browser, ma ad oggi nessuna implementazione in tal senso è ancora stata avviata e quindi risulta ancora prematuro dare dei giudizi definitivi riuscendo a valutare se le API fornite da NW.js forniscano tutte le caratteristiche necessarie per far fronte alle richieste di progetto, si rimanderà quindi tale valutazione ad un lavoro successivo.

Componente intermedio di navigazione

Come ampiamente mostrato nei capitoli precedenti, lo scenario relativo al forensics-browser ha presentato carenze sulla gestione della sessione SSL/TLS e relativa estrazione delle chiavi necessarie a decifrare il traffico di rete acquisito, ma allo stesso tempo si è mostrato come l'estrazione delle suddette chiavi sia un'attività possibile e implementabile con relativamente poche righe di codice.

Seppure NW.js possa potenzialmente risolvere il problema, ad oggi non c'è una reale implementazione che possa dare piena conferma del pieno supporto alle necessità di progetto, risulta quindi fondamentale avere una strategia alternativa che possa arginare il problema e che permetta eventualmente di riciclare il lavoro svolto facendo uso di

Electron. Per far fronte a tale necessità si è quindi fatto strada la necessità di introdurre un componente aggiuntivo che potesse fungere da ruolo di mediatore tra il forensics-browser e la navigazione. A tutti gli effetti il componente svolgerebbe un ruolo di "http proxy" in grado di rimuovere la cifratura SSL/TLS prima di passarla al browser, ma così catalogato sarebbe fuorviante e con il relativo rischio di indurre in cattive interpretazioni.

Il componente sarebbe a tutti gli effetti un agente software eseguito con le credenziali dell'operatore e il modo corretto di interpretare la sua funzione è considerarlo come un componente esterno del browser in vita sulla stessa stanza locale in cui viene eseguito il browser stesso, permettendo connessioni solamente ad esso e solamente su interfaccia di loopback (127.0.0.1). Quindi seppure un agente software autonomo il suo ruolo è strettamente legato a quello del forensics-browser, e nulla vieterebbe di cablarlo nel medesimo codice del browser, ma per motivi di modularità e manutenibilità è preferibile "esternalizzarlo".

A fronte dell'aggiunta di questo modulo, anche il ruolo del browser risulterebbe alterato, rendendolo di fatto non più il componente principale di navigazione ma una "semplice" interfaccia grafica in grado di permettere all'utente di pilotare proprio il comportamento del componente esterno, il quale risulterà quindi essere il vero e proprio mediatore della connessione in grado sia di tenere traccia di tutte le risorse caricate e della loro relativa acquisizione che di estrapolare le eventuali chiavi SSL/TLS.

Eventuali critiche sulla non ortodossa pratica forense che incolpi il componente di effettuare modifiche al contenuto veicolato sono facilmente contestabili indicando come un componente simile potrebbe essere implementato internamente al browser e che una qualsiasi altra linea di codice interna all'applicativo potrebbe fare quanto contestato e se così fosse, risulterebbe ce qualsiasi programma non dovrebbe essere considerato affidabile. Inoltre si fa nuovamente presente come il modulo esterno debba accettare solamente connessioni sull'interfaccia locale e solo dal browser associato, per fare ciò il browser potrebbe inviare un token autenticativo sempre differente e prestabilito ad ogni nuova esecuzione della sessione di acquisizione. [Figura 25]

Dumper

Il componente dumper ha il ruolo di svolgere il dump fisico del traffico di rete. Così da poter tracciare le connessioni tra client e server individuando oltre alle singole richieste HTTP anche le richieste DNS e i relativi indirizzi IP contattati. Questo componente

risulta quindi fondamentale per dare coerenza e attendibilità a quanto acquisito dal forensics browser.

Questo componente risulta essere formato da uno script bash eseguito ad inizio sessione. Al suo interno si trovano le necessarie direttive da passare a *iptables* in grado di permettere un "ragruppamento" delle connessioni per UID (User ID) di interesse. Risulterà quindi possibile l'intercettazione con il comando *dumpcap* usufruendo del relativo parametro utile a intercettare il gruppo appena creato. Una volta eseguito dumpcap con le sopra indicate impostazioni sarà intercettato tutto il traffico trasmesso e ricevuto da ogni processo eseguito dall'utente identificato dall'UID, e quindi salvato su file.

Di seguito le righe necessarie per svolgere quanto necessario su traffico proveniente da un ipotetico utente avente uid 1000:

```
$ iptables -A OUTPUT -m owner --uid-owner 1000 -j CONNMARK --set-mark 1
$ iptables -A INPUT -m connmark --mark 1 -j NFLOG --nflog-group 30
$ iptables -A OUTPUT -m connmark --mark 1 -j NFLOG --nflog-group 30
$ dumpcap -i nflog:30 -w uid-1000.pcap
```

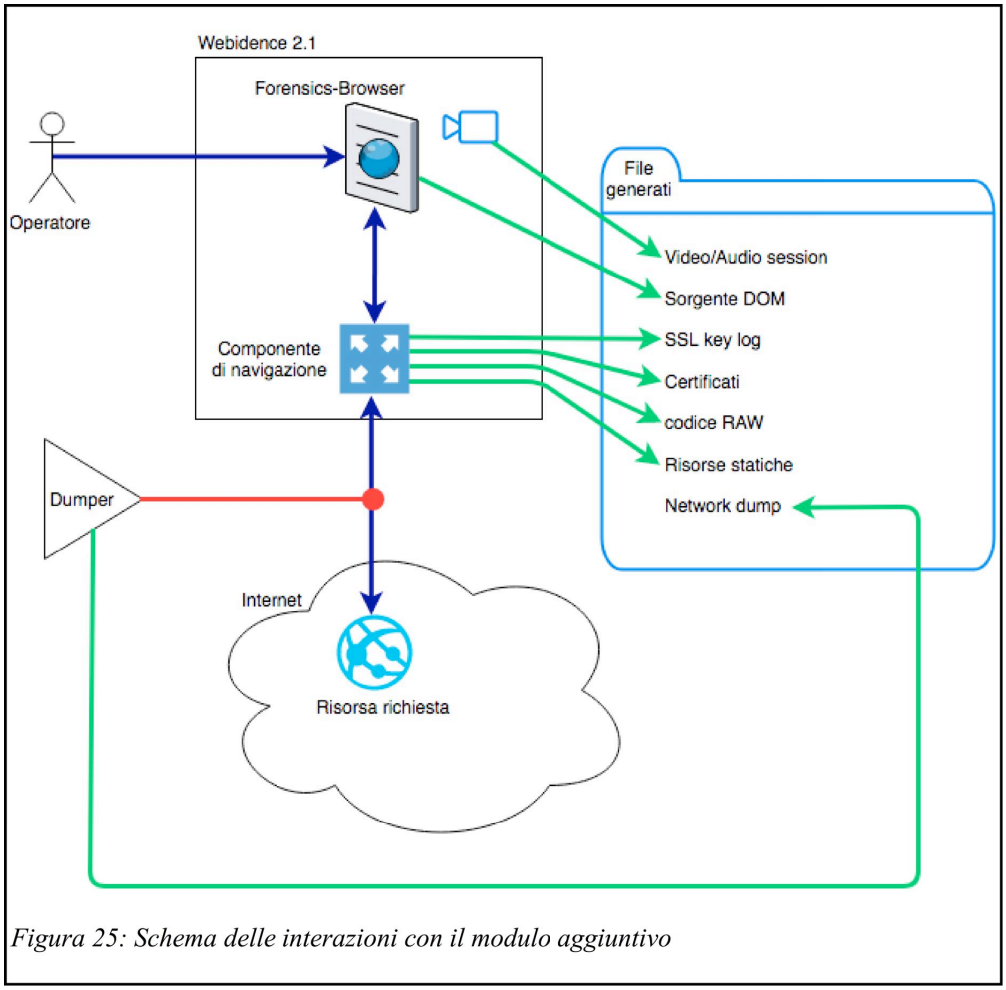


Figura 25: Schema delle interazioni con il modulo aggiuntivo

Futuri sviluppi

Pur con tanto materiale prodotto, il progetto è sicuramente ben lungi dall'essere completato, presentando evidenti mancanze in attesa di essere ancora implementate:

- riscrivere il forensics-browser basandosi sul frame-work NW.js anziché Electron, sfruttando l'occasione per implementare le parti ad oggi ancora mancanti nella versione fin qui realizzata
- implementazione del dialogo tra frontend e backend in grado di instaurare il canale di comunicazione necessario a trasferire la sessione Xpra verso il browser dell'operatore

Nello specifico, il primo punto che prevede la riscrittura del forensics-browser con NW.js dovrà prevedere la possibilità che tale re-implementazione possa far emergere l'inadeguatezza di NW.js a far fronte alle necessità richieste e se così fosse risulterà quindi necessario passare alla realizzazione del componente intermedio di navigazione in grado di svolgere la funzione di SSL/TLS inspection così da poterne estrapolare le chiavi di sessione.

Pur a fronte delle predette mancanze si può comunque pacificamente affermare che l'attività di ricerca e sviluppo finora svolto può essere considerato pienamente sufficiente per decretare la solida fattibilità degli obiettivi posti dalla presente tesi

Sviluppi futuri non trattati precedentemente nel presente documento saranno:

- applicare la marca temporale al report generato facendo uso di una authority riconosciuta,
- predisporre una retention dell'acquisizione generata su server certificati che ne possano garantire la genuina coerenza di integrità nel tempo

A proposito dei punti appena esposti si è individuato in “Poste Italiane” il potenziale partner tecnologico più adatto, in quanto in grado di soddisfare pienamente i servizi richiesti e di renderli fruibili remotamente tramite il protocollo SOAP. Poste Italiane ha inoltre l'importante vantaggio (in ottica nazionale) di svolgere solidamente il ruolo di garante riconosciuto.

Conclusioni

Dal presente elaborato si è potuto rilevare come al giorno d'oggi, anche a fronte di decenni a contatto con le moderne tecnologie digitali, la società sia in un certo senso "*disorientata*" (spesso inconsapevolmente) da questa stretta convivenza. Convivenza che ha permesso di creare una vera e propria simbiosi tra la "*mediasfera*" e la singola personale "*noosfera*".

In questo moderno (e bizzarro) rapporto tra "uomo e macchina" non mancano di certo degli strani contrasti; da un lato vi è sempre più grande dimestichezza tecnica nell'utilizzo dei vari dispositivi (computer, smartphone, smart-TV, droni, etc), dall'altro questa confidenza non va però di pari passo con la consapevolezza. La gente fa molta fatica ad interiorizzare quanto i nuovi media stiano influenzando la società, sia negli usi e costumi, ma ancor di più da un punto di vista giuridico.

Le persone presentano una cognizione troppo "astratta" del "mondo digitale", trovando non poche difficoltà a considerarlo "estensione **reale e non virtuale** della propria realtà sociale". Questa **inconsapevolezza** ha però delle inevitabili ripercussioni molto concrete, in quanto un utilizzo errato (o spesso malsano) dei dispositivi digitali porta con sé inevitabili conseguenze legali.

La giurisprudenza di merito, anche se con molta fatica e in maniera non sempre completa, ha cercato e continua a tentare di dare risposte a queste nuove evoluzioni, apportando nuove leggi o modificando quelle già presenti.

Oltre all'analisi del contesto antropologico e forense, si sono inoltre individuate "categorie" legislative di rilievo, così da identificare al meglio i campi applicativi di interesse nei quali dovrà operare il software sviluppato.

Successivamente, si è spostata l'attenzione sugli aspetti tecnologici; prima di tutto sono state analizzate le soluzioni "concorrenti" ad oggi presenti, così da poter identificare al meglio lo stato dell'arte in essere; in seconda istanza si è proceduto alla vera e propria fase di sviluppo del progetto, che ha visto la realizzazione sia di codice originale che delle dovute configurazioni utili all'integrazione con altre soluzioni necessarie a creare l'ambiente adeguato.

Dall'analisi dei software con intenzionalità similari, si è potuto riscontrare la presenza di un ampio ventaglio di strumenti in grado di generare snapshot di contenuti Web. Tra queste, ad oggi soltanto **tre** (tutte italiane) presentano funzionalità forensi, e soltanto **due** di queste in grado di gestire contenuti Web moderni (complessi e asincroni):

- **HashBot** (https://www.hashbot.com/) è un servizio OnLine ideato da Gianni Amato. È in grado di effettuare l'acquisizione di una singola risorsa fruibile tramite HTTP/s. L'output generato conterrà oltre alla risorsa di interesse anche la firma hash di quanto acquisito, riferimenti temporali e log HTTP. La sua più grande **carenza** è il suo stesso scopo: acquisizione di una singola risorsa statica; risulta quindi impossibile, per sua stessa definizione di prodotto, poter acquisire contenuti Web complessi.
- **FAW** (http://www.fawproject.com/) è un software per Windows ideato da Davide Bassani e Matteo Zavattari, sviluppato in .NET unitamente al framework CEF (Chromium Embedded Framework). È un "browser forense" che permette di effettuare acquisizioni di contenuti Web complessi. Sono state individuate le seguenti **carenze**: impossibilità di decifrare il traffico HTTPS acquisito, assenza del codice raw, mono piattaforma, assenza tra i log degli header HTTP delle singole risorse e dei redirect.
- **LegalEYE** (https://www.legaleye.cloud/) è un software ideato dalla LegalEYE srl di Udine. Viene reso fruibile come servizio Web tramite piattaforma Citrix e sistema virtualizzato basato su Windows. Al suo interno viene eseguita una versione "personalizzata" di Mozilla Firefox che ha il compito di generare l'acquisizione dei contenuti. Sono state individuate le seguenti **carenze**: viene tutto demandato al "salva con nome" di Firefox, assenza del dump di rete, assenza tra i log degli header HTTP delle singole risorse e dei redirect.

La fase di sviluppo ha visto la realizzazione di due soluzioni distinte. Questa necessità è emersa da problemi architetturali riscontrati durante la fase realizzativa della prima versione, capaci di introdurre criticità tali da non rendere possibile non solo il raggiungimento degli obiettivi prefissati, ma anche quelli che progressivamente si introducevano con la maggiore consapevolezza indotta dall'attività di sviluppo e l'inevitabile evoluzione del progetto stesso.

La **prima fase**, denominata con il nome in codice "Webidence 1.0", consisteva in un'applicazione completamente Web based avente l'obiettivo di frapporsi tra le richieste dell'utente e il server remoto ospitante la risorsa di interesse. Risultava essere un "Web Proxy" con funzionalità aggiuntive, tali da permettere all'utente di eseguire richieste di acquisizione per la risorsa in esame. Vi erano inoltre funzionalità nascoste che, lavorando in background, permettevano di rendere fruibili tutti quei contenuti che introducevano barriere di accesso più o meno marcate.

Pur essendo riusciti a rendere funzionante questa prima versione, essa presentava non pochi limiti e criticità, tali da renderla inadatta allo scopo. Più precisamente, i limiti della soluzione erano i seguenti: impossibilità di catturare il dump del traffico di rete, acquisizione posticipata con relativa incertezza della presenza del dato di interesse, limiti del singolo browser nel permettere e gestire i video capture della sessione di lavoro, necessità di alterare i sorgenti così da poter fornire al client un contenuto completamente fruibile, non piena estraneità del prodotto da componenti di terze parti potenzialmente presenti sulle macchine degli utenti, come per esempio anti-virus, ad-blocker, firewall (fisici o software), add-ons varie, … etc.

Ultima criticità (ma non per questo meno importante), instabilità del prodotto a fronte di perturbazioni esterne non controllabili, in grado di rompere la compatibilità con determinati servizi (es: Facebook); questa situazione rendeva ingestibile la manutenibilità, in quanto richiedeva una costante verifica di compatibilità e conseguente rimessa mano al codice del modulo di interesse. Situazione che si poteva verificare, nei casi peggiori, anche più volte al giorno, rendendo la sua gestione insostenibile sia economicamente che operativamente.

La **seconda fase** è nata dalle ceneri della prima e ha visto un repentino cambio di tecnologia, passando da un un'applicazione Web based ad un'applicazione sviluppata con tecnologie "*desktop based*". In tale fase è stato sviluppato un vero e proprio "Browser con finalità forensi", reso fruibile attraverso un normale browser Web tramite una sua renderizzazione server side e veicolata al client tramite "Web Socket".

Questa seconda versione, che ha preso il nome in codice di “Webidence 2.0” , è riuscita a risolvere tutte quelle criticità presenti nella prima fase esposte precedentemente.

Tra i risultati importanti di questa seconda versione si annovera la possibilità di effettuare il dump del traffico di rete puntualmente per ogni singola istanza, potendone **interpretare successivamente anche il traffico HTTPS** in esso presente grazie all'intercettazione delle chiavi **SSL/TLS** client e server. Questo aspetto risulta particolarmente importante poiché, senza questa possibilità, l'intercettazione del traffico di rete risulterebbe solo un ammasso di “blobware” per lo più inutile, escludendo l'identificazione degli indirizzi IP contattati e le richieste DNS effettuate.

“Webidence 2.0” è risultato quindi essere un ibrido tra FAW e LegalEye, in grado di cogliere il meglio delle due soluzioni oltre ad aggiungere proprie specificità originali. Si è quindi coniugato lo sviluppo di un vero e proprio “Forensics Browser” che permetta all'operatore una fruizione per mezzo di un comune Web browser (che supporti le Web Socket) senza la necessità di installare nulla sui propri sistemi.

In conclusione a tutto il lavoro di studio e sviluppo svolto espresso nel presente documento, si è giunti ad un'accurata **identificazione delle necessità che un moderno software di acquisizione forense per contenuti Web deve offrire**. Nello specifico:

- sorgente RAW del codice
- informazioni *whois* di tutti gli IP e domini contattati
- intercettazione delle modifiche attuate al DOM nelle varie elaborazioni client e relativa estrapolazione del codice finale al momento della finalizzazione
- singole risorse collegate alla pagina (css, immagini, video, script, pagine annidate in frame, etc)
- intercettazione di tutti gli eventi dinamici avvenuto nella pagina(esecuzione di script e relative funzioni, modifiche del DOM)
- log del traffico asincrono intercorso
- log di tutti gli header HTTP presenti, sia per il contenuto principale di interesse che di tutte le risorse ad esso collegate
- log di tutti i redirect necessari a raggiungere sia il contenuto di interesse che le singole risorse
- acquisizione dei certificati SSL/TLS
- dump del traffico di rete

- estrazione delle chiavi SSL/TLS client e server necessarie a decifrare il contenuto HTTPS
- registrazione della sessione audio/video
- screenshot del contenuto, sia a richiesta dell'utente che automatico
- (opzionale) salvataggio delle risorse acquisite anche in formato WARC - "Web ARChive"(https://www.loc.gov/preservation/digital/formats/fdd/fdd000236.shtml)

Futuri sviluppi:

Ad oggi la soluzione si presenta come un insieme di mattoncini non completamente integrati tra loro, risultando quindi essere una versione dalle rosee potenzialità ma ancora non completamente espresse. Compito principale per il prossimo futuro sarà quindi quello di amalgamare armoniosamente quanto fino ad oggi costruito.

Per permettere agli utenti di utilizzare il tutto come un servizio Web, sarà inoltre necessario creare l'infrastruttura necessaria a consentire la comunicazione tra server di frontend e server di backend, così da permettere una sicura ed efficiente integrazione e interazione tra lo strumento operativo e l'area esposta pubblicamente.

Per lo sviluppo del "forensics browser" si è fatto uso di Electron (http://electron.atom.io/), ma per necessità di copione risulterà necessario effettuare un test di migrazione sul framework NW.js (https://nwjs.io/), così da poter sfruttare il supporto nativo (ma non documentato) di intercettazione delle chiavi SSL/TLS. Nel caso NW.js non risultasse idoneo sotto aspetti ad oggi non prevedibili, sarà necessario proseguire nell'utilizzo dell'attuale "forensics browser" sviluppato con Electron. Ma l'utilizzo dell'attuale forensics browser, per limiti del framework Electron, espone la necessità di aggiungere un componente esterno (ad oggi sviluppato solo come proof of concept) in grado di intercettare le chiavi SSL/TLS. Risulta invece scartata la possibilità di sviluppare il Forensics Browser utilizzando CEFPython + override delle connessioni con UrlLib3-PyOpenSSL. Questa soluzione, pur funzionante come proof of concept, è risultata essere troppo azzardata, con il rischio non remoto di presentare comportamenti instabili e imprevedibili.

Risulterà altresì necessario implementare la generazione di un report completo e sopratutto marcato temporalmente da un'authority riconosciuta (es: Poste Italiane), così da associare data e ora certe e legalmente valide, consentendo quindi di associare una

validazione temporale opponibile a terzi[2]. (cfr. Art. 20, comma 3 Codice dell'Amministrazione Digitale Dlgs 82/2005).

Oltre a quanto appena esposto, risulterà utile ai fini delle attività investigative integrare tra gli strumenti disponibili anche la possibilità di acquisire le singole "*entità social*" nella loro totalità (profili, gruppi, pagine o altre nomenclature simili presenti sui vari social network). A tal proposito alcuni studi preliminari sono già stati fatti e, per quanto riguarda l'implementazione, essa prevederà la scrittura distinta di uno specifico modulo separato da quanto esposto fino ad ora. L'attività di acquisizione completa di un'entità social prevede nuove complicazioni che necessitano di mettere in strada molteplici strategie che prevederanno sia di usare le API ufficiali che di fare attività di Web Scraping tra versioni desktop, mobile o ultra-mobile. Visti i tempi necessari e svolgere l'attività di acquisizione completa di un'entità social, questa sarà probabilmente gestita come task schedulato ed eseguito in background e non come acquisizione immediata.

Pur a fronte delle predette mancanze si può comunque pacificamente affermare che l'attività di ricerca e sviluppo finora svolto può essere considerato pienamente sufficiente per decretare la solida fattibilità degli obiettivi posti dalla presente tesi

2 *È l'idoneità di un atto giuridico ad esprimere la sua efficacia anche nei confronti dei terzi e non solo delle parti. Diversamente, l'atto inopponibile ha effetti limitati esclusivamente alle parti che vi hanno dato vita.*

Ringraziamenti

Dopo mesi di intenso operato in una commistione tra studio e lavoro, finalmente è arrivato il cortese momento di scrivere le dovute frasi di ringraziamento verso chi, con grande pazienza, mi è stato vicino dandomi supporto in questo cammino.

Il primo ringraziamento va al mio relatore, il **Professore Fabio Vitali**, per i suoi preziosi consigli.

Al mio stimato correlatore aziendale, **dott. Stefano Fratepietro** che ha preso anche lui sulle spalle questo importante impegno, fornendomi le dovute risorse e disponibilità per poter operare al meglio e portarlo a termine con successo.

Al mio qualificato quanto apprezzato consulente e referente "legal", **dott. Niccolò Baldi**, che con grande competenza e pazienza è stato in grado di fornirmi i giusti spunti su temi a me così lontani come quelli giuridici e di propormi inoltre le giuste migliorie da apportare alla forma oltre che alla sostanza.

Vorrei inoltre ringraziare **tutti i ragazzi di Tesla** per la loro fantastica pazienza e il loro supporto morale. Mi hanno sempre sostenuto e sono sempre stati pronti ad aiutarmi.

Un ringraziamento ulteriore va alla mia amica **dott.sa Amanda Accalai** che silentemente ci ha sempre creduto.

Un grande ringraziamento a **mia madre e mio padre** che, con il loro dolce e instancabile sostegno mi hanno permesso di arrivare fin qui davanti a voi oggi, contribuendo alla mia formazione personale.

GRAZIE A [A-Z][a-z]+

Massimiliano Dal Cero

Bibliografia

[GL2012]	I reati informatici: Disciplina sostanziale e questioni processuali - Gianluca D'Aiuto . Luigi Levita - Giuffrè Editore - 2012
[VC2017]	Vania Contraffatto - Reati informatici - Key Editore - 2017
[RF2001]	Ruggiero Francescopaolo - Ciberspazio E diritto penale il problema del bene giuridico -Rivista Penale Volume 127 - 2001
[CA2006]	Calice Andrea - I Computer crimes nell'ordinamento giuridico italiano - 2006
[SB2014]	Stefano Sbordoni - Web, libertà e diritto - 2014
[CE2016]	CENSIS Il fatturato della contraffazione vale 6,9 miliardi di euro - http://www.censis.it/7?shadow_comunicato_stampa=121067 - 2016
[CS2006]	Mario Centorrino, Olimpia Scopelliti, Enciclopedia Italiana – VII Appendice. Contraffazione dei Prodotti, 2006.
[CCCD2016]	Le nuove frontiere dell'acquisizione degli elementi di prova nel cyberspace - F. CAJANI, G. CERNUTO, G. COSTABILE, F. D'ARCANGELO - iisfa Forensics Associaation - 2016
[FG2013]	Digital Forensics - Gabriele Faggioli, Andrea Ghirardini – Apogeo - 2013
[U1]	Il mondo virtuale non esiste - Consapevolezza Digitale http://consapevolezzadigitale.org/blog-post/57-il-mondo-virtuale-non-esiste

[RS2012]	Presi dalla Rete: La mente ai tempi del Web - Raffaele Simone - Garzanti - 2012
[PT1995]	Pierre Teilhard de Chardin (1955) - Il fenomeno umano, Queriniana, Brescia, 1995
[ZF2011]	Liar, Liar, Hard Drive on Fire: How Media Context Affects Lying Behavior - Zimpler & Feldman (2011). http://onlinelibrary.wiley.com/doi/10.1111/j.1559-1816.2011.00827.x/abstract - 2011
[IQ2013]	Internet e l'Io diviso: La consapevolezza di sé nel mondo digitale - Ivo Quartiroli - 2013
[DQ2004]	DECISIONE QUADRO 2004/68/GAI http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2004:013:0044:0048:IT:PDF
[AG2011]	"definizione e principi giuridici dell'informatica forense" – Avv. Antonio Gammarota – Università di Catania
[CM2012]	"Ruolo e prospettive dell'Informatica Forense"- Dott. Cesare Maioli
[CM2012b]	I "nuovi"" mezzi di ricerca della prova fra informatica forense e L. 48/2008 - 07/05/2012
[EC2011]	E. Casey, Digital evidence and computer crime, Elsevier, 2011
[CM2004]	C. Maioli, Dar voce alle prove: elementi di informatica forense - 2004
[LZ2007]	L. Lupària, G. Ziccardi Investigazione penale e tecnologia informatica, Giuffrè, 2007

[GC2006]	G.Costabile, Ecco come procedono al sequestro di un pc, Punto informatico del 23/05/2006 http://punto-informatico.it/1494286_3/PI/News/ecco-come-procedono-al-sequestro-del-pc.aspx
[BBC2017]	http://www.bbc.com/news/world-39729819
[NG2016]	Группы смерти (Gruppy Smerti = Gruppi di morte) https://www.novayagazeta.ru/articles/2016/05/16/68604-gruppy-smerti-18
[PS2017]	“Blue Whale: i consigli della Polizia postale” - Polizia di Stato: http://www.poliziadistato.it/articolo/3859303d1ed8e5a566574486
[WPBW]	https://en.wikipedia.org/wiki/Blue_Whale_(game)
[CP15060]	Cassazione penale, sez. V 23/02/2011 n. 15060
[TR2007]	Trib. Roma, Sez. IX, 30 ottobre 2007, n. 22615, in Il Merito , Speciale n. 1, 2008, 49.
[TM2010]	Trib. di Monza Sezione IV Civile, sentenza 2 marzo 2010, n. 770,
[TM2011]	Trib. Milano, 31 marzo 2011, in Resp. Civ. Prev. , n. 6, 2011, 1320 ss.
[AL2014]	Reato di molestie: a che condizioni Facebook è da considerarsi luogo pubblico? Cassazione penale, sez. I, sentenza 12/09/2014 n° 37596 http://www.altalex.com/documents/news/2014/09/22/reato-di-molestie-a-che-condizioni-facebook-e-da-considerarsi-luogo-pubblico
[CP2010]	Cass. Pen., Sez. V, 5 luglio 2010, n. 25527

[PS2016]	Cyberstalking: le nuove frontiere del diritto penale , Pramila Sicuro - 2016 https://www.diritto.it/cyberstalking-le-nuove-frontiere-del-diritto-penale/
[Art .612-bis CP 2009]	*art. 612-bis c.p. D.L. 23.2.2009, n. 11,*
[AL2015]	Altalex - Dl anti-terrorismo: in Gazzetta la legge di conversione – 2015 http://www.altalex.com/documents/leggi/2015/04/15/dl-anti-terrorismo-ok-senato-alla-fiducia-e-legge-con-161-si-e-108-voti-contrari
[MI2016]	Diritto e nuove tecnologie – Michele Iaselli – Altalex Professionale - 2016
[TL2012]	Tribunale Livorno, ufficio GIP, sentenza 31.12.2012 n° 38912
[LP2012]	Lorenzo Picotti,I diritti fondamentali nell'uso ed abuso dei social network. Aspetti penali, Giurisprudenza di Merito, fasc.12, 2012, pag. 2522B
[AL2017]	Cyberbullismo: la legge pubblicata in Gazzetta – Altalex – 2017 http://www.altalex.com/documents/news/2016/09/21/bullismo-e-cyberbullismo 18 giugno 2017
[LM1993]	Legge Mancino - legge 25 giugno 1993, n. 205
[L131975]	Legge n.13 ottobre 1975 , n.654

[PB2013]	“OPERAZIONE HOLYWAR” della DIGOS Oscuramento del sito antisemita “holywar.org” e perquisizioni in varie città italiane http://questure.poliziadistato.it/it/Bolzano/articolo/5730de9982c29431270744 18 aprile 2013
[L201952]	Legge 20 giugno 1952, n. 645
[SR2016]	Modifiche alla Legge 20 giugno 1952, n. 645, sulla produzione, distribuzione, diffusione e vendita di beni mobili raffiguranti immagini o simbologie del disciolto partito fascista http://www.senato.it/japp/bgt/showdoc/17/DDLPRES/964870/index.html?stampa=si&spart=si&toc=no 21 gennaio 2016
[CS2017]	Cass. sent. n. 24103/17 del 15.03.2017
[WKXPRA]	https://en.wikipedia.org/wiki/Xpra ultima visita 13 settembre 2017
[EL2017]	https://electron.atom.io/ ultima visita 31 ottobre 2017
[ED2017]	Electron Documentation https://electron.atom.io/docs/development/atom-shell-vs-node-webkit/ ultima visita 31 ottobre 2017

Indice delle figure