



Alcune vulnerabilità della Rete risalgono ai primi protocolli di comunicazione. Come si può migliorare la sicurezza?

bisogno *reale* dello stesso protocollo impiegato per richiedere e trasmettere le pagine di un sito Web (anche se spesso lo usiamo *anche* per quei servizi). I problemi del Web attuale sono un ottimo esempio di quelli generali di tutta Internet, soprattutto perché parte delle soluzioni, che descriveremo in queste pagine, sono le stesse.

Traffico Internet più sicuro: se non ora, quando?

Se dobbiamo trasferirci tutti su Internet, non solo per chiacchiere su Facebook o leggere qualche notizia ogni tanto, ma per farne il nostro mezzo *principale* di studio, lavoro, comunicazioni, rapporti con le Pubbliche Amministrazioni, qualcosa deve cambiare presto, sia nella Rete sia nel nostro modo di usarla. Questo mese parliamo di una delle cose più importanti che dovrebbero cambiare, degli strumenti Open Source per farlo e del perché questo sia il momento adatto.

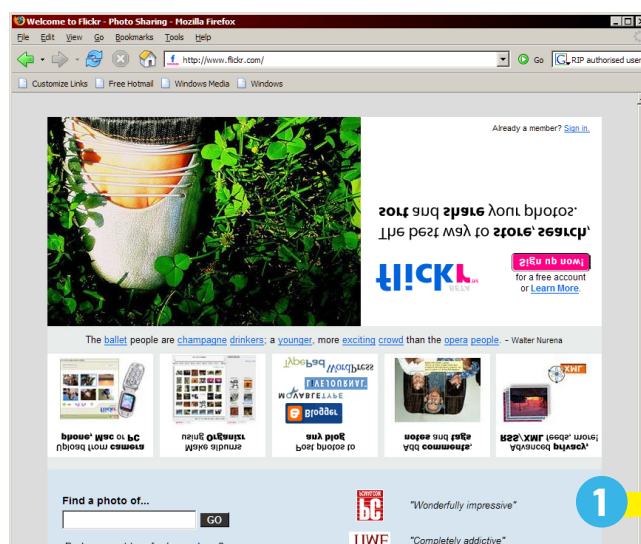
HTTP, LA BASE DEL WEB

Internet è un insieme di protocolli di comunicazione e infrastrutture globali che rendono intercomunicanti tutte le reti "locali" del mondo (anche se coprono intere nazioni), trasformandole in un unico sistema di telecomunicazioni globale. Il World Wide Web, invece, è solo una *piccola* parte di Internet, anche se molte persone ancora pensano che i due termini (o addirittura Facebook, che è solo una minima parte del Web) siano la stessa cosa. Usando una definizione, non rigorosa ma efficace, il Web è l'insieme di contenuti e servizi fruibile attraverso

browser come Firefox, Chrome o Explorer. Per capire come questa non sia che *una* delle applicazioni di Internet, basta pensare a trasferimento di file, messaggistica istantanea, email, streaming video, telefonia con Skype o simili. Questi e tanti altri servizi essenziali, per non parlare delle connessioni private dirette fra sedi diverse di una stessa organizzazione, girano su Internet ma non hanno alcun

Il protocollo usato da browser e server di siti Web per scambiarsi pagine e altri contenuti si chiama Http (*Hyper Text Transport Protocol*). Il suo primo serio problema, anche se non l'unico ed è lo stesso di tutta la Internet originaria, è che non è cifrato. Chiunque riesca a intercettare il traffico Http fra browser e server può immediatamente "rubare" nomi di account e relative password, cookie, testi di email o qualsiasi altro dato, incluso l'elenco completo di

Alterare in tempo reale le pagine Web non cifrate, come in questo esempio in cui tutte le immagini sono capovolte, è facilissimo con il software giusto e l'accesso a un qualunque router posto fra sito e utente.



tutte le pagine visualizzate. Questa debolezza può essere utilizzata, da chiunque, da Stati a criminali comuni; e lo è, come ci insegnano il caso Snowden e i continui resoconti di furto d'identità online. L'unica cosa buona di questa vulnerabilità, senza dubbio grave anche se troppi utenti sembrano ancora non preoccuparsene affatto, è che almeno è di comprensione immediata.

L'alterazione di traffico in tempo reale però è ancora più pericolosa, proprio perché è meno facile rendersi conto di cosa può avvenire *in pratica*. Il modo più efficace di spiegarlo è forse la Figura 1, una schermata d'esempio di Upside-Down-Ternet (Udt), letteralmente "Internet capovolta" (www.ex-parrot.com/pete/upside-down-ternet.html): collegandosi a un sito attraverso un qualsiasi router o server configurato come quello originale di Udt, quest'ultimo ruoterà automaticamente tutti i file grafici richiesti dal browser, prima di inoltrarli.

Di per sé, Udt è solo uno scherzo, o al massimo un dispetto senza serie conseguenze. Il problema, di cui Udt è solo un'ottima dimostrazione pratica, è un altro. La tecnica che adotta è molto facile da utilizzare, a portata di qualsiasi amministratore di sistema che abbia accesso ai router o computer su cui passa il traffico; ed è, soprattutto, *la stessa* tecnica con cui si potrebbero riscrivere al volo intere pagine, o addirittura falsificare interi siti. Tutti i nostri computer e smartphone riescono a connettersi a un qualsiasi sito

Web, come www.esempio.com, soltanto chiedendo ad appositi server l'indirizzo numerico corrispondente a ogni nome di dominio. Riscrivendo in corsa le risposte di quei server, cosa tecnicamente quasi identica al rivoltare le immagini, diventa quindi possibile far connettere l'utente a un sito fasullo. Una copia perfetta di quello vero, *incluso* l'indirizzo che appare nell'apposita barra del browser!

Il protocollo Http, in sostanza, non è soltanto *troppo trasparente*, cioè incapace di celare ai malintenzionati qualsiasi dato riservato che dovesse trasmettere: è anche privo di difese contro i falsari. Chiunque usi servizi su Web, di qualsiasi tipo, o qualsiasi applicazione di Internet con le stesse limitazioni di base, non può avere alcuna garanzia certa che all'altro capo di una connessione ci sia veramente chi dice di esserci.

TRE SIGLE FONDAMENTALI: HTTPS, SSL E TLS

La soluzione già praticabile, in moltissimi casi, per i problemi appena descritti sta nell'usare connessioni cosiddette Https: la S finale indica la sicurezza (S, appunto) aggiunta ad Http dai protocolli Ssl/Tls, che forniscono sia cifratura sia meccanismi di autenticazione dei siti. Usati correttamente, questi sistemi aumentano notevolmente la riservatezza e soprattutto l'integrità (cioè la protezione da corruzioni dei dati e falsificazioni) delle comunicazioni su Internet.

VENT'ANNI FA, I PRIMI CERTIFICATI DI GARANZIA

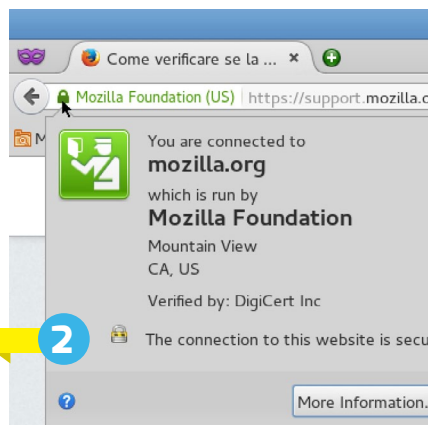
La prima versione della cifratura che è alla base di Https arrivò nel 1994, con l'invenzione di Ssl (*Secure Socket Layer*) da parte di Netscape. I certificati Ssl a esso collegati, invece, sono documenti che autenticano in modo univoco un certo dominio e il server Web che lo ospita: insieme, protocollo e certificati garantiscono che si sta accendendo davvero, e in maniera riservata, al sito che si desidera e non ad altri.

Nel 1998 il progetto OpenSSL (www.openssl.org) mise a disposizione di tutti, per praticamente qualsiasi sistema operativo, una versione interamente Open Source di Ssl, e in seguito del suo

CHI E COME PUÒ SPIARCI SU WEB?

Usando connessioni non sicure come quelle Http

o simili, chiunque può intercettare il traffico per copiare dati riservati oppure sostituirsi ai siti con cui vorremmo effettivamente parlare. Ma chi può farlo, in pratica? Beh, tanto per cominciare, almeno tutti gli amministratori (di diritto o di fatto, cioè cracker, poco importa) dei vari access point WiFi che usiamo da smartphone o laptop, in qualunque ambiente: dai vicini di casa ad aeroporti, uffici e biblioteche, il discorso è sempre lo stesso. Tecnicamente, è ovvio, gli stessi attacchi sono possibili anche attraverso connessioni cablate, perché il problema è il protocollo Http, non quelli di networking. Quelle connessioni sono relativamente più sicure di quelle WiFi solo perché, in molti casi, è più difficile per i malintenzionati riuscire a controllarne i router corrispondenti. Ci sono poi connessioni, soprattutto wireless ma non solo, che potrebbero produrre "assuefazione" pericolosa: quelle dei punti di accesso a Internet, cablati o no, che presentano di prepotenza una qualche pagina di login quando ci si collega la prima volta, qualunque fosse il sito Web che si voleva raggiungere. Questo è il caso tipico di hotel, aeroporti e Internet Café che offrono connettività a pagamento, o comunque solo dopo essersi identificati con una qualche organizzazione. Per obbligare l'utente a effettuare quella procedura (e a volte anche per inserire pubblicità nelle pagine visitate dopo di essa) si usano le stesse tecniche appena descritte: il traffico Internet in ingresso viene bloccato e sostituito fino a nuovo ordine con il modulo di login. Tutto questo è inevitabile, ma ha un effetto collaterale di cui occorre essere consapevoli: navigare attraverso router configurati in quel modo può confondere i browser, portandoli a lanciare abbastanza all'armi ingiustificati per "connessioni non sicure", da indurre l'utente a ignorare anche quelli veri. Mai abbassare la guardia, quindi.



Il pulsante di identificazione di Firefox mostra in maniera molto semplice quanto ci si può fidare sia della connessione a un sito, sia della sua identità.

successore. L'anno successivo, infatti, quel protocollo venne rimpiazzato da quello chiamato Tls (*Transport Layer Security*). Anche se ancora oggi, soprattutto per inerzia, spessissimo si dice Ssl anziché Tls. Quando sono supportati entrambi, Ssl/Tls, è Tls che gestisce tutte le connessioni più sicure per Web, posta elettronica e numerosi altri servizi. La parte più complicata di Tls non è la cifratura vera e propria quanto i documenti che abbiamo già menzionato, veri e propri certificati d'identità che un browser scarica automaticamente dai siti a cui stiamo per connetterci.

Questi certificati possono essere "autofirmati", cioè creati dagli stessi webmaster di un sito, oppure validati con diversi gradi di garanzia da apposite società, chiamate *Certificate Authority* (Ca). I certificati autofirmati, per ovvie ragioni, sono quelli che causano i vari avvertimenti di "sito non attendibile" in tutti i browser. Quelli più garantiti, in verde nella *Figura 2* (vedi Box Risorse), vengono rilasciati solo dopo varie verifiche e accordo firmato ("Extended Validation") tra richiedente e Certificate Authority.

HTTPS? SÌ, MA PER BENE

Perché le connessioni via Https risolvano davvero tutti i problemi per cui sono state ideate, occorre che ogni server usi solo versioni *correnti* dei vari protocolli, ovvero quelle più recenti di Tls vero e proprio, insieme a chiavi crittografiche della massima dimensione possibile. Subito dopo, sempre per garantire la massima protezione ai propri utenti, in teoria bisognerebbe evitare senza eccezioni di servire pagine che, per funzionare correttamente, carichino fogli di stile, script, immagini o video... via link non Https.

Facendo così, i problemi che abbiamo descritto rimarrebbero tali e quali, limitandosi a entrare nel computer da ingressi secondari, anziché da quello principale. Lo stesso discorso vale per i cookie. Anche loro dovrebbero essere validi solo per domini accessibili via Https e viaggiare solo su connessioni cifrate.

Sarebbe opportuno passare prima possibile anche all'estensione del protocollo Http chiamata Hsts (*Strict-Transport-Security Header*, <https://https.cio.gov/hsts/>): usandola, un sito Web può imporre a tutti i browser di usare sempre

e solo Https per le comunicazioni e di rifiutare certificati sospetti, anche se l'utente volesse farlo, o se chiedesse deliberatamente connessioni non cifrate, scrivendo "http://", senza "S" finale, nella barra degli indirizzi.

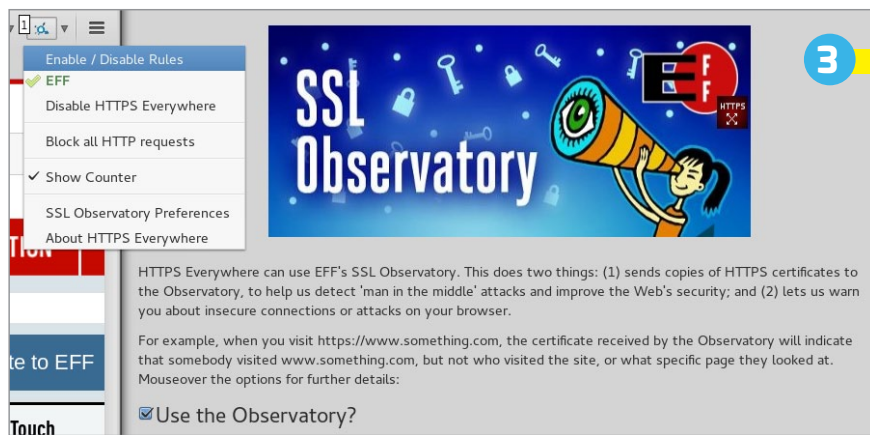
In pratica, fino a oggi, fare tutto questo è stato molto più facile a dirsi che a farsi. A meno di non essere webmaster molto esperti, con molto tempo e spazio Web a disposizione, sarebbe molto difficile seguire alla lettera questi consigli con software come WordPress, Drupal o Joomla, magari su account di hosting a basso costo. In casi del genere, a parte il fatto che non sarebbe possibile cambiare la configurazione del server, si dovrebbero modificare i temi grafici e usare pochi o nessun plugin, riducendo molto la funzionalità di un sito. Nonostante questo, anzi *proprio* per questo, rimane essenziale per chiunque voglia gestire un sito Web con software Open Source (e non solo) seguire il "mercato", per capire quali prodotti porteranno queste funzioni alla portata di tutti prima degli altri. Nell'attesa, è comunque possibile fare molto per accelerare la transizione a un Web "100% Https".

LE POSSIBILITÀ DEGLI UTENTI?

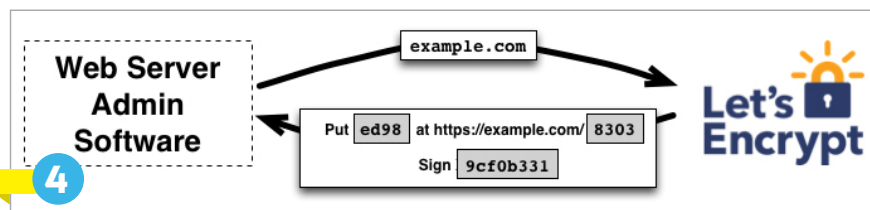
Il modo più semplice e immediato per arrivare prima possibile a un Web sicuro è pretendere, come utenti, solo siti e servizi online interamente cifrati, nel senso appena descritto. Per verificare quanto fidarsi dell'identità di un sito, dovremmo tutti usare sistematicamente sistemi come il pulsante di identificazione di Firefox di *Figura 2*: fatelo, e rimarrete sorpresi da quanti siti anche critici, di tutti i generi, non utilizzano i migliori certificati possibili.

La richiesta di passare completamente a Https dovrebbe valere anche nel settore in gran crescita degli e-book: senza cifrature e certificati adeguati, con procedure standard, sia i libri elettronici, sia i relativi sistemi di abbonamento, consultazione e acquisto online possono essere vulnerabili tanto quanto i siti Web tradizionali.

Un sistema davvero alla portata di tutti per visitare correttamente tutti i siti Web che già sfruttano al massimo Https è l'Https Everywhere di *Figura 3* (www.eff.org/HTTPS-everywhere), che d'ora in poi chiameremo HttpE per semplicità. Questo piccolo, ma utilissimo pezzo di software Open Source è un'estensione per browser, lanciata alcuni anni fa da



La finestra di configurazione e (a sinistra) il menu di HttpE, con cui è facile evitare connessioni insicure e contribuire a un censimento di tutti i siti che già utilizzano (o no) i certificati Ssl.



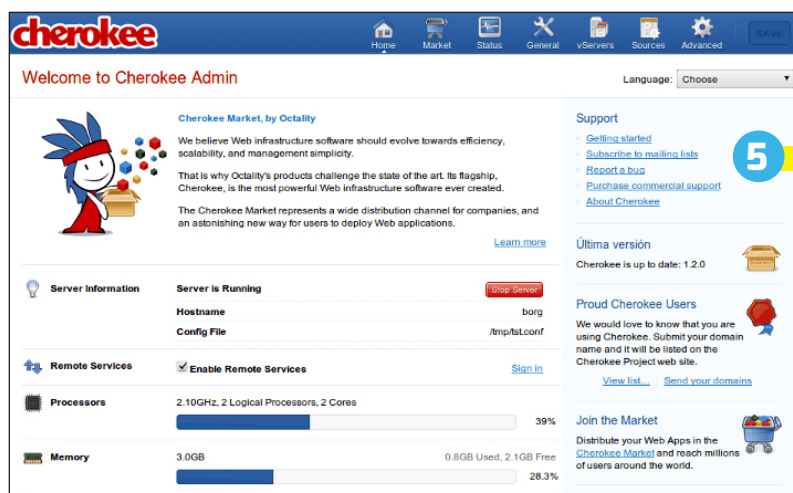
Let's Encrypt verificherà automaticamente l'identità dei server che le richiedono certificati Ssl, chiedendogli di pubblicare sui loro siti messaggi opportunamente cifrati.

Electronic Frontier Foundation (www.eff.org) e progetto Tor (www.torproject.org) e disponibile dall'anno scorso anche su Android. Il suo effetto pratico è ancora limitato, se paragonato all'immensità del Web, ma già molto utile: una volta installata, HttpE capisce se il sito che si vuol visitare utilizza pienamente Https, e costringe il browser a connettersi esclusivamente in modalità cifrata. Questo avviene grazie ai cosiddetti *ruleset* che si installano insieme al software vero e proprio di HttpE: elenchi di siti Web 100% Https, completi delle istruzioni per utilizzarli *solo* in quel modo.

SSL/TLS, PROSSIMA PUNTATA

Se Ssl/Tls è così utile, perché solo una minoranza di siti Web lo usa al 100%? I motivi principali sono l'impatto della cifratura sui server, e le difficoltà di ottenere e gestire certificati di qualità. In realtà la prima ragione, innegabile fino a pochi anni fa, è sempre meno valida e per la seconda sono previste grosse novità, ovviamente Open Source, entro l'estate del 2015. Per quanto riguarda le prestazioni, praticamente tutti i siti Web moderni sono molto più limitati dalla velocità dei loro database interni e dalla banda delle loro connessioni che dalla cifratura Https.

A questo si può rimediare in parte riducendo il numero dei collegamenti fra server e browser necessari per visualizzare una pagina completa, e quindi la quantità di calcoli da fare. L'esempio migliore di questa affermazione, anche non è certo uno immediatamente replicabile da programmatori indipendenti, è quello di Gmail. Secondo i suoi sviluppatori, Https contribuisce a meno



Cifratura, autenticazione e molte altre funzioni di un sito Web sono interamente configurabili da interfacce grafiche con server come Cherokee.

dell'uno per cento del carico sulle Cpu dei server Gmail, e questo risultato è dovuto in buona parte proprio a tecniche come l'accorpamento di più richieste da browser a server in una sola. Passando ai certificati, i loro costi attuali (sia economici, sia burocratici e gestionali) sono dovuti principalmente alla complessità e ai costi delle poche Certificate Authority che finora hanno monopolizzato questo mercato. Perché la validazione via Ssl/Tls funzioni davvero come previsto, e con le massime garanzie, un browser ha infatti bisogno di sapere quali Ca possono confermare con certezza l'autenticità dei certificati che richiede a un sito. Le liste delle Ca davvero affidabili, normalmente chiamate "root store", sono incluse in tutti i browser e anche in vari sistemi operativi. Per semplificare l'uso dei certificati stessi, sono quindi

necessarie due cose. Una è affiancare alle Authority attuali organizzazioni egualmente affidabili sul piano tecnico, ma con costi di gestione estremamente più bassi. L'altra è convincere i fornitori di browser e sistemi operativi ad aggiungerle ai loro "root store" ufficiali. Il progetto più promettente, interamente Open Source, è quello chiamato Let's Encrypt (<https://letsencrypt.org>) di Eff, Mozilla, Cisco e altre organizzazioni: entro l'estate 2015 Let's Encrypt offrirà, tramite procedure automatizzate di cui un passo è mostrato nella figura 4, certificati gratuiti e facili da configurare, insieme a software per installarli sul proprio server in meno di un minuto. Con un minimo di fortuna, il prossimo anno non ci saranno più scuse per non offrire siti Web, email e altri servizi in modalità sicure e autenticate. •



RISORSE

Il funzionamento del pulsante di identificazione di Firefox ha una sua guida ufficiale (<https://support.mozilla.org/it/kb/verificare-connessione-sito-web-sicura>). Le risorse migliori per configurare correttamente Https e Tls, con uno qualunque dei server Web descritti questo mese, sono senz'altro le guide e/o mailing list in Italiano su questi argomenti, fornite dalle specifiche distribuzioni Linux su cui si volesse far girare un sito Web sicuro. Eff ha pubblicato una guida dettagliata su come scrivere ruleset per Https Everywhere (<https://www.eff.org/https-everywhere/rulesets>), insieme a un Atlante di tutti i siti già riconosciuti e serviti da questa estensione (www.eff.org/https-everywhere/atlas). Oltre ai webmaster, l'Atlante è utile anche a qualsiasi utente che volesse sapere, prima di connettersi, se un certo sito sarà visitabile in maniera sicura grazie a quel plugin per browser. Chiunque trovi un errore nei ruleset, o volesse proporli di nuovi, può scrivere alla mailing list pubblica https-everywhere-rules@eff.org. La documentazione completa di Let's Encrypt, invece, è stata pubblicata su <https://letsencrypt.readthedocs.org/en/latest>, mentre le procedure di generazione delle chiavi, da cui viene la figura 4, si trovano su <https://letsencrypt.org/howitworks/technology>.



Il server più diffuso è Apache, ma diversi suoi concorrenti possono offrire livelli di sicurezza uguali o superiori e maggiore flessibilità.

Https sì ma... con quale server?

Oggi, su Linux e non solo, installare un server Web con supporto completo Https non è un problema. La prima difficoltà, in effetti, potrebbe essere *scegliere* quale server utilizzare, fra i vari prodotti Open Source disponibili. La risposta più ovvia potrebbe essere il server Apache (<http://apache.org>), che da vent'anni manda avanti la maggioranza dei siti di tutto il mondo. L'estrema versatilità di Apache è però anche il motivo per cui in diversi casi la miglior scelta possibile potrebbe non essere lui, ma uno qualunque dei suoi concorrenti Open Source presentati nei prossimi paragrafi.

CHEROKEE

WWW.CHEROKEE-PROJECT.COM

Un web server completo per Linux e OS X, il cui punto di forza è la facilità di gestione. Tutta la configurazione avviene infatti attraverso Cherokee-Admin, un'interfaccia grafica accessibile con qualsiasi browser. Non c'è alcun bisogno, se non lo si desidera, di scrivere a mano i file di configurazione, o di conoscerne la sintassi esatta. Oltre a questo, Cherokee-Admin include parecchi "assistenti" per tutti i servizi più comuni, dall'interprete Php a vari sistemi di gestione dei contenuti. Sono questi assistenti che provvedono da soli, appena caricati, a modificare almeno parte della configurazione base per poter utilizzare il servizio corrispondente. Sempre con assistenti, o comunque sempre da interfaccia grafica, si possono attivare più server virtuali su un unico computer, gestire l'autenticazione degli utenti, mostrare con grafici il carico del

server ed eventualmente distribuirlo su più macchine virtuali. Il tutto senza dover riavviare Cherokee.

HIAWATHA

WWW.HIAWATHA-WEBSEVER.ORG

Hiawatha sembra particolarmente adatto a chi vuole gestire siti Web dinamici, ma bloccando con poco sforzo, sul nascere, le cause più comuni sia di attacchi informatici, sia di sovraccarichi del server. Questo software può riconoscere e neutralizzare, per esempio, le cosiddette "iniezioni Sql", ovvero l'invio di comandi che potrebbero cancellare, in tutto o in parte, il database interno di un sito. Altre procedure provvedono al bilanciamento del carico, a ottimizzare l'invio di file di grandi dimensioni e a limitare il tempo di esecuzione di varie applicazioni, proprio per evitare blocchi del server.

LIGHTTPD

WWW.LIGHTTPD.NET/

Questo server si distingue dagli altri in queste righe soprattutto per una cosa: è il più adatto, secondo molti dei suoi utenti, per i microcomputer come Raspberry Pi (www.raspberrypi.org) e Beagle Board (<http://beagleboard.org>). In altre parole, Lighttpd sembra la scelta migliore per un media center personali da salotto, o condivisione online di gallerie di foto digitali senza affidarsi a servizi esterni come Flickr, ma usando hardware Open Source che costa pochi Euro, consuma pochissimi Watt e occupa

pochi centimetri quadrati. La configurazione di Lighttpd è relativamente semplice anche se, certamente, molto meno di quella di Cherokee. Il suo amministratore può impostare, fra le altre cose, server virtuali, limitazioni dell'impatto sulla Cpu, compressione del traffico, autenticazione e pagine Web dinamiche con Php sempre in *un solo* file di testo, anziché parecchi come avviene con Apache. Altra grande caratteristica di questo server è il supporto per WebDNA (<http://webdna.us>): un linguaggio di scripting per creare siti Web dinamici compatibili con tutte le moderne tecnologie Web, da Ajax a Css3 e Html5.

NGINX

HTTP://NGINX.ORG

Subito dopo Apache, anche se con un gran distacco, nginx è il server Web Open Source più popolare. Prestazioni e affidabilità sono testimoniate dal fatto che se ne servono portali con fatturati e/o volumi di traffico altissimi, da Netflix e Pinterest a Wordpress.com e AirBnB. Nginx è installabile come pacchetto binario su tutte le distribuzioni Linux più importanti, ma gira anche su Bsd, Mac OS X e Windows. Chi lo usa lo fa soprattutto perché, oltre a essere più veloce di Apache in molti scenari pratici, è anche molto più "leggero" e affidabile, nel senso che il consumo di memoria all'aumentare del traffico cresce di meno, e in maniera più prevedibile, di quello del suo "cugino" superstar.

Il server più facile
Grazie a una serie di assistenti per i servizi più comuni, l'alternativa più semplice ad Apache è Cherokee

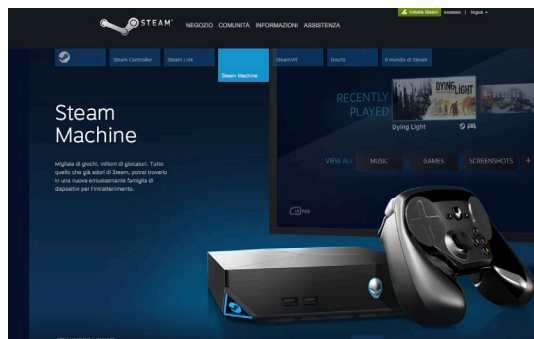
NEWS

FIREFOX, UN ASSISTENTE SEMPRE PIÙ VERSATILE

Le versioni di Firefox disponibili da fine estate 2015 conterranno due funzioni capaci di trasformare questo browser in un assistente tuttotfare ancora più versatile, sia per lo studio sia per il lavoro. Il servizio di video chat Firefox Hello (www.mozilla.org/it/firefox/hello) guadagna la possibilità di condividere una scheda del browser, o la finestra di qualsiasi programma, durante le conversazioni con altri utenti di Hello. Su tutt'altro fronte, aumenta l'integrazione con Pocket (<https://getpocket.com>), un'interfaccia per creare e condividere, fra tutti i propri computer, liste di pagine Web o video da consultare in un secondo momento. Anche offline, perché Pocket permette di fare copie locali dei contenuti selezionati.



ARRIVANO LE STEAM MACHINES



Steam Machine (<http://store.steampowered.com/universe/machines>) è il nome di una famiglia di computer assemblati e configurati, secondo le specifiche di Valve Software (www.valvesoftware.com), appositamente per giocare con gli oltre mille videogame disponibili sulla sua piattaforma online chiamata Steam. Le prime Steam Machine, costruite da Alienware e CyberPower, dovrebbero arrivare sul mercato a novembre 2015. Su tutte loro sarà installata la distribuzione Linux appositamente creata per questo scopo, chiamata SteamOs (<http://store.steampowered.com/steam/>), che contiene il client con cui scaricare i giochi. SteamOs può essere installata direttamente dagli utenti su computer autocostruiti, ma ovviamente le Steam Machines renderanno disponibile questa piattaforma anche a chi non ha tempo o conoscenze per farlo. Inoltre, secondo Valve, dalle Machines sarà possibile giocare anche con i titoli per Windows, se in rete locale è disponibile un altro computer con questo sistema operativo.

LINUX, ARDUINO, RASPBERRY PI E FPGA INSIEME, IN UN UNICO PACCHETTO

Arduino (<http://arduino.cc>) è il microcontroller Open Source più famoso del mondo, alla base di migliaia di progetti dall'hobbistica di tutti i tipi al monitoraggio ambientale. Raspberry Pi (www.raspberrypi.org) e Beagle Board (<http://beagleboard.org>) sono microcomputer su cui è possibile far girare diverse distribuzioni Linux. Il primo è particolarmente popolare perché costa poche decine di Euro ed è grande quanto una carta di credito. Le Fpga (Field Programmable Gate Array) sono circuiti integrati digitali in cui i collegamenti fra transistor, ovvero le loro stesse funzioni hardware, sono programmabili in modo completamente diverso a ogni accensione, secondo i desideri dell'utente. Da giugno 2015 due schede dotate di Fpga della serie ValentFX di Newark Element (<http://valentfx.com>) chiamate Logi-Pi e Logi-Bone, consentono di connettere ancora più facilmente tutte queste piattaforme, per creare microcomputer completamente personalizzati e capaci sia di calcoli complessi sia di interazioni con l'ambiente esterno, tramite i sensori e relay direttamente controllabili da Arduino. Oltre all'integrazione puramente hardware, infatti, la piattaforma ValentFX offre anche tre diversi ambienti di sviluppo, chiamati complessivamente "Logi-Ecosystem", per programmare sia le Fpga sia gli altri elementi hardware nel modo più semplice possibile.

