

Crypto-Gram  
15 novembre 2001  
Scritto da Bruce Schneier  
Fondatore e CTO di Counterpane Internet Security, Inc.

Edizione italiana curata da Communication Valley

e-mail: [schneier@counterpane.com](mailto:schneier@counterpane.com)

CRYPTO-GRAM è una newsletter mensile gratuita che offre riassunti, analisi, approfondimenti e commenti sulla sicurezza informatica e sulla crittografia.

\*\* \*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

In questo numero:

[Esposizione totale](#)  
[Le ristampe di Crypto-Gram](#)  
[News](#)  
[Le News di Counterpane Internet Security](#)  
[GOVNET](#)  
[Vulnerabilità di Password Safe](#)  
[Microsoft a riguardo di Windows XP](#)  
[Commenti dei lettori](#)

\*\* \*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

Esposizione totale

Microsoft è alla guida della causa contro la diffusione di informazioni che riguardano i buchi di sicurezza in ambito informatico. Lo scorso mese Scott Culp, capo del centro della sicurezza informatica di Microsoft, ha pubblicato un articolo che descrive la pratica corrente di rendere note informazioni sulle vulnerabilità come "anarchia d'informazione". Egli sostiene che saremmo tutti più al sicuro se i ricercatori tenessero per sé i dettagli riguardanti i buchi di sicurezza, smettendo così di offrire mezzi d'attacco per gli hacker. La scorsa settimana, al Trusted Computing Forum, Culp ha annunciato una nuova coalizione per mettere in pratica tutte queste idee.

Questa è la classica querelle "segretezza sui bug contro esposizione totale". Ne ho già parlato precedentemente su Crypto-Gram; anche altri ne hanno parlato. Si tratta di una problematica complicata, con sottili implicazioni riguardanti l'intero discorso della sicurezza informatica, e vale la pena discuterne ancora.

La Finestra di Esposizione

Ho coniato il termine "Finestra di Esposizione" ("Window of Exposure") per spiegare l'evoluzione di una falla di sicurezza nel tempo. Una vulnerabilità è un bug; è un errore di programmazione fatto da chi ha scritto il programma durante lo sviluppo del prodotto e non lo ha riscontrato durante la fase di test. Si tratta di un'apertura che qualcuno può sfruttare per penetrare all'interno di un sistema o per fare qualcosa in genere vietato.

Poniamo che esista una vulnerabilità in un prodotto e nessuno ne sia al corrente. Il pericolo è quasi nullo, perché nessuno sa come forzare la vulnerabilità. Questa "falla" può rimanere

nascosta per un breve periodo (le falle di Windows XP furono scoperte ancor prima del lancio del prodotto), oppure per anni. Alla fine qualcuno scopre questa vulnerabilità. Forse si tratta di una brava persona che avverte gli sviluppatori. Forse si tratta di un malintenzionato che sfrutta questa vulnerabilità per forzare qualche sistema. Forse si tratta di qualcuno che non avverte nessuno, e in seguito qualcun altro, dopo mesi magari, la scopre a sua volta. In ogni caso, nel momento in cui qualcuno è a conoscenza di questa vulnerabilità, il pericolo aumenta.

Alla fine, la notizia di questa falla si diffonde. Può diffondersi all'interno della comunità della sicurezza informatica. Così come può diffondersi all'interno delle comunità hacker. Il pericolo aumenta con l'aumentare del numero di persone a conoscenza di questa vulnerabilità. Ad un certo punto la vulnerabilità viene annunciata: può darsi che sia annunciata su Bugtraq o su un altro sito Web che parla di buchi di sicurezza. Può darsi che sia il ricercatore stesso ad annunciarla in un comunicato stampa, oppure che venga annunciata dal CERT o dallo sviluppatore del software stesso. Può anche darsi che venga annunciata su un forum di discussione di hacker. In ogni caso, una volta che la vulnerabilità è annunciata, il pericolo aumenta ancora di più, perché ancora più persone ne sono a conoscenza.

Poi qualcuno scrive un'exploit: uno strumento automatizzato che sfrutta la vulnerabilità. Questo è un punto di flesso, che non ha un corrispettivo nel mondo reale, per due ragioni. In primo luogo il software ha la capacità di separare la capacità dall'abilità. Una volta che uno strumento viene scritto, chiunque è in grado di forzare la vulnerabilità, a prescindere dalla propria bravura o comprensione del mezzo. In secondo luogo, questo strumento può venire ampiamente distribuito a costo zero, offrendo così le capacità a chi le desidera. A questo punto entrano in gioco gli "script kiddies", i ragazzini che giocano a fare gli hacker, persone che utilizzano strumenti di attacco automatizzati per penetrare all'interno di sistemi. Una volta che viene scritto uno strumento di forzatura, il pericolo aumenta esponenzialmente.

In seguito lo sviluppatore del software pubblica una patch. Il pericolo diminuisce, ma non così tanto quanto ci piacerebbe credere. Un gran numero di computer in Internet non hanno le patch aggiornate: vi sono moltissimi esempi di sistemi violati sfruttando falle di sicurezza che avrebbero dovuto essere riparate da una patch. Non dò colpa ai vari amministratori di rete, esistono tante patch e molte di queste sono scritte male e testate peggio. Così, anche se il pericolo decresce, non ritornerà mai a livello zero.

Si può pensare a questo come ad un diagramma pericolo-tempo, e alla Finestra di Esposizione come all'area sottostante il diagramma. Il fine ultimo è quello di ridurre quest'area quanto più possibile. In altre parole vogliamo che ci sia il minor pericolo possibile durante il ciclo vitale di un software e della specifica vulnerabilità. I sostenitori della segretezza dei bug da una parte, e quelli dell'esposizione totale dall'altra, hanno semplicemente idee diverse per raggiungere quest'obiettivo.

### Storia dell'Esposizione Totale

Agli albori dell'era informatica e delle reti, la segretezza sui bug era la norma. Quando utenti e ricercatori scoprivano delle vulnerabilità in un prodotto software, essi avvertivano il produttore con discrezione. Il produttore si sarebbe poi occupato, almeno in teoria, di sistemare la vulnerabilità. Nel 1988 venne fondato il CERT (Computer Emergency Response Team) che divenne immediatamente un punto di raccolta per quanto concerne le vulnerabilità. Gli utenti mandavano al CERT le falle di sicurezza appena scoperte, il CERT le verificava, avvertiva i produttori, e poi pubblicava i dettagli della vulnerabilità nonché la patch una volta che essa fosse stata disponibile.

Il problema di questo metodo stava nel fatto che i produttori non avevano alcun interesse a riparare le vulnerabilità. Il CERT non avrebbe pubblicato nulla prima dell'arrivo di una soluzione, e dunque non c'era urgenza. Era molto più semplice tenere nascoste le vulnerabilità. C'erano casi di produttori che avrebbero minacciato i ricercatori se questi avessero reso pubbliche le loro scoperte, e campagne diffamatorie nei confronti di quei ricercatori che

annunciavano l'esistenza di vulnerabilità (anche se non entravano nei dettagli). E così parecchie vulnerabilità rimanevano tali per anni.

Il movimento per l'esposizione totale nacque dalla frustrazione che questo processo generò. Una volta che una vulnerabilità viene resa nota, le pressioni del pubblico danno ai produttori un forte incentivo per sistemare il problema rapidamente. Nella maggior parte dei casi questo sistema ha funzionato. Oggi molti ricercatori rendono note le vulnerabilità da loro scoperte attraverso mailing list come Bugtraq. La stampa scrive di queste vulnerabilità su riviste specializzate. I produttori si affannano per realizzare delle patch a queste vulnerabilità una volta che esse vengono diffuse, in modo che possano poi scrivere i loro comunicati stampa vantandosi di come siano stati rapidi ed efficaci nel sistemare le cose. Il movimento per l'esposizione totale sta facendo aumentare la sicurezza in Internet.

Allo stesso tempo, gli hacker utilizzano quelle mailing list per studiare le vulnerabilità e scrivere poi degli exploit. A volte gli stessi ricercatori scrivono degli exploit dimostrativi. Altre volte sono altre persone a scriverli. Questi exploit vengono usati per penetrare in computer e reti vulnerabili, e fanno diminuire parecchio la sicurezza informatica. Nel suo articolo, Culp cita Code Red, LiOn, Sadmin, Ramen e Nimda quali esempi di codice scritto con intenti illeciti dopo che i ricercatori hanno spiegato come funzionavano certe vulnerabilità.

Coloro che si oppongono al movimento per l'esposizione totale sostengono che pubblicare dettagli riguardanti le vulnerabilità procura più male che bene, poiché mette nelle mani degli hacker strumenti che possono usare per penetrare nei sistemi. Al contrario essi sostengono che la sicurezza informatica può trarre giovamento dal mantenere segreti i dettagli specifici delle varie vulnerabilità.

I sostenitori dell'esposizione totale ribattono che questo implica che il primo ricercatore che diffonde notizie su una vulnerabilità è colui che la scopre, e ciò non è affatto vero. A volte certe vulnerabilità erano note ai malintenzionati (perché tranquillamente diffuse nell'underground hacker) per mesi o anni prima che il produttore ne venisse a conoscenza. Prima una falla di sicurezza viene resa nota e riparata, meglio è per tutti, essi sostengono, e ritornare alla segretezza sui bug non farebbe altro che riportarci al menefreghismo dei produttori e all'inerzia.

Questo è, molto sinteticamente, il succo della questione: il beneficio di pubblicizzare un attacco vale la minaccia sempre più concreta che un nemico sia in grado di sfruttare questa informazione? Dovremmo forse ridurre la Finestra di Esposizione cercando di limitare la conoscenza della vulnerabilità, oppure rendendo nota questa vulnerabilità per costringere i produttori a sistemarla quanto prima?

Ciò che abbiamo visto negli ultimi otto anni è che l'esposizione totale produce più benefici che danni. Dato che essa è diventata la norma, l'industria informatica si è trasformata: da un gruppo di aziende che ignora le questioni di sicurezza e minimizza i problemi legati alle vulnerabilità, in un sistema che ripara queste vulnerabilità il più presto possibile. Alcune aziende vanno addirittura oltre, prendendo molto seriamente la questione della sicurezza e cercando di realizzare prodotti di buona qualità sin dall'inizio, sistemando le vulnerabilità prima del rilascio del prodotto. Pochi e remoti problemi stanno diventando importanti nelle comunità hacker, che attacca le persone senza alcun preavviso. Una volta le informazioni riguardanti le vulnerabilità erano appannaggio di una ristretta cerchia, quei ricercatori e quegli hacker più esperti nei loro rispettivi campi. Ora quelle informazioni sono alla portata di tutti.

Questa democratizzazione è importante. Se una vulnerabilità viene resa nota e voi non lo sapete, allora state attuando delle decisioni sulla sicurezza usando informazioni insufficienti. La notizia si diffonderà - la Finestra di Esposizione crescerà - ma non siete in grado di tenere la cosa sotto controllo, né avete idea di quando e come ciò avverrà. Tutto quel che potete fare è sperare che i cattivi non scoprano il problema prima che i buoni lo risolvano. L'esposizione

totale significa che tutti ricevono le informazioni allo stesso momento, e che tutti possono agire di conseguenza.

Inoltre servono informazioni dettagliate. Se un ricercatore pubblica semplicemente delle frasi vaghe a riguardo di una certa vulnerabilità, allora il produttore può rispondere che non è vero. Se un ricercatore pubblica dettagli tecnici senza esempi di codice, allora il produttore può rispondere che si tratta soltanto di teoria. L'unico modo per smuovere e far agire i produttori è quello di pubblicare informazioni dettagliate, sia in formato cartaceo sia elettronico. (Microsoft è abituato a ciò, dato che sfrutta il proprio apparato di pubbliche relazioni per negare e minimizzare le vulnerabilità finché non vengano dimostrate efficacemente con parti di codice). Il codice dimostrativo è il solo modo di verificare che la patch prodotta per riparare una vulnerabilità sia davvero efficace.

Questo flusso di libera informazione, sia di codice descrittivo che di codice "a prova di concetto", è vitale per la ricerca sulla sicurezza informatica. Ricerca e sviluppo, in questo ambito, sono fioriti nell'ultima decade, e molto si deve al movimento per l'esposizione totale. La possibilità di pubblicare scoperte di ricerca - sia buone sia cattive - porta ad una maggiore sicurezza per tutti. Senza le pubblicazioni, la comunità che si occupa di sicurezza informatica non può imparare dai reciproci errori. Tutti finirebbero col lavorare coi paraocchi e finirebbero col ripetere sempre gli stessi sbagli. L'esposizione totale è essenziale se vogliamo migliorare la sicurezza dei nostri sistemi e delle nostre reti.

#### Un esempio di segretezza sui bug

I problemi creati dalla segretezza sui bug si possono notare nell'industria che gestisce i diritti digitali. Il DMCA ha gelosamente conservato il paradigma della segretezza sui bug all'interno della legge; nella maggior parte dei casi è illegale rendere note vulnerabilità o tool di hack ai danni di schemi anti-copia. Vengono fatte pressioni sui ricercatori, affinché non distribuiscano il loro lavoro. Le falle di sicurezza vengono tenute segrete. Come risultato si ha una moltitudine di sistemi deboli e poco sicuri, e i loro proprietari inveiscono contro la legge sperando che nessuno scopra quanto deboli sono i loro sistemi.

La conseguenza di tutto ciò è che gli utenti non riescono a prendere decisioni intelligenti riguardanti la sicurezza. Eccone un esempio: qualche mese fa, il ricercatore per la sicurezza Niels Ferguson scoprì un buco nel sistema di crittografia video digitale HDCP di Intel, ma non ha rivelato i dettagli poiché teme di venire perseguito in base al Digital Millennium Copyright Act (DMCA). Intel ha reagito nel perfetto stile del periodo pre-esposizione totale, sostenendo che la forzatura è stata solo "teorica" e che il sistema era ancora sicuro. Immaginate di dover valutare l'acquisto del sistema Intel. Che fare? Non siete in possesso di vere e proprie informazioni, così dovrete fidarvi o di Ferguson o di Intel.

Ecco un altro esempio: alcune settimane fa è stata rilasciata una versione del kernel Linux senza l'abituale documentazione dettagliata riguardante la sicurezza del sistema operativo. Gli sviluppatori hanno giustificato la reticenza dichiarando timori inerenti al DMCA. Immaginate di star provando diversi sistemi operativi: vi sentite più o meno tranquilli per quanto concerne la sicurezza del kernel Linux 2.2 ora che non avete informazioni specifiche?

#### Esposizione totale e responsabilità

Culp non ha torto quando parla di responsabilità (naturalmente Scott sta evitando un "mea culpa"). L'obiettivo è quello di aumentare la sicurezza, non di armare persone che penetrano all'interno di sistemi e di reti. Tools di hack, dotati di interfacce intuitive, già pronti per i vari "script kiddies" provocano parecchi danni alle organizzazioni e alle loro reti. Si può distinguere fra esposizione responsabile ed irresponsabile. Delineare le differenze non è sempre facile, ma credo di avere alcune linee guida.

Innanzitutto sono contro quegli attacchi che principalmente seminano paura. Pubblicare vulnerabilità senza avere prove concrete è negativo. Pubblicare vulnerabilità che sono tutto fumo e niente arrosto è negativo. Pubblicare vulnerabilità di sistemi critici che non possono essere riparate facilmente e la cui forzatura potrebbe causare danni gravi (ad esempio: il sistema di controllo del traffico aereo) è negativo.

Secondariamente, ritengo sia utile che il produttore sia informato anticipatamente. Il CERT portò questa pratica all'estremo, a volte dando al produttore anni interi per sistemare il problema. Mi piacerebbe che il ricercatore informasse il produttore che la vulnerabilità verrà resa nota nel giro di alcune settimane, e che poi mantenesse la promessa. Attualmente il CERT dà ai produttori 45 giorni di tempo, ma diffonde istantaneamente agli abbonati le informazioni sulla vulnerabilità. Microsoft propone un periodo di segretezza di 30 giorni. Se da un punto di vista teorico questa è una buona idea, creare una élite di persone "che sanno" porta a tutta una serie di problemi.

In terzo luogo, sono d'accordo con Culp quando afferma che è irresponsabile e in fondo criminale distribuire exploit facili da usare. Fare del reverse engineering su sistemi di sicurezza, scoprire vulnerabilità, scrivere articoli di ricerca su di esse e persino scrivere codice dimostrativo è di beneficio alla ricerca, e rende più capaci a progettare sistemi sicuri. Distribuire exploit rende soltanto più deboli. Vorrei mettere le mani su quelli che distribuiscono kit di creazione di virus, per esempio. Avrebbero molte cose di cui rispondere.

Questo non è ben delineato: esistono strumenti che fanno cose buone e cattive, e spesso la differenza è puramente questione di marketing. Dan Farmer è stato diffamato per aver scritto SATAN; gli strumenti per misurare le vulnerabilità sono prodotti di amministrazione della sicurezza che danno profitti. Gli strumenti di amministrazione remota assomigliano molto a Back Orifice (anche se hanno meno funzioni). LOphtCrack è uno strumento degli hacker per violare password non sicure ed è la fase iniziale di un attacco, ma allo stesso tempo LC 3.0 viene venduto come prodotto di amministrazione di rete che serve a testare password non sicure. Il programma per cui Dmitry Sklyarov è stato arrestato ha degli usi legittimi. Infatti molti strumenti possono avere un utilizzo buono o cattivo e quando sorgono dubbi io ritengo sia meglio dare le informazioni necessarie a chi ne ha bisogno, sebbene questo significhi che le possano ottenere anche i malintenzionati.

Una cosa da tener presente è l'intento del ricercatore. Rendere nota una vulnerabilità è spesso un gioco di pubblicità; il ricercatore sta cercando di veder pubblicato il proprio nome sui giornali come ricompensa per aver acchiappato la sua preda. Spesso anche il pubblicitario ha i propri intenti; è un consulente per la sicurezza, oppure un impiegato di una compagnia che offre prodotti e servizi per la sicurezza informatica. Sono un po' stufo di quelle compagnie che pubblicano vulnerabilità con lo scopo di spingere i propri prodotti o servizi. D'altro canto delle motivazioni non altruistiche non significano necessariamente cattive informazioni.

Mi piace la filosofia di "essere parte della soluzione e non parte del problema". Fare ricerche nell'ambito della sicurezza è parte della soluzione. Convincere i produttori a sistemare i problemi è parte della soluzione. Seminare paura è parte del problema. Offrire strumenti di attacco a ragazzini ignoranti è parte del problema.

L'inevitabilità dei buchi di sicurezza

Nulla di quanto detto sarebbe un problema se il software fosse scritto in maniera appropriata già in partenza. Un buco di sicurezza è un errore di programmazione, sia esso un errore bello e buono come l'overflow di un buffer, che avrebbe dovuto essere intercettato ed evitato, sia esso un'apertura introdotta dalla mancanza di comprensione delle interazioni in un pezzo di codice particolarmente complesso. Se non ci fossero falle di sicurezza, non ci sarebbero problemi. La prima causa di questi pasticci è la scarsa qualità del software.

Se da un lato, purtroppo, questo è vero - che si produce e si vende software di scarsa qualità - il grado di complessità del software e delle reti oggi è tale da rendere inevitabili molte vulnerabilità. Esse si trovano in qualsiasi grande package. Ogni volta che Microsoft rilascia un sistema operativo si vanta di quanto estesi siano stati i test e di quanto sicuro il prodotto sia, e puntualmente esso contiene più vulnerabilità del sistema precedente. Non credo che questo andamento cambierà presto.

I produttori non prendono molto sul serio la questione della sicurezza perché non c'è alcun incentivo di mercato in questo senso, e non ci sono effetti sfavorevoli. Ho a lungo sostenuto che i produttori di software non dovrebbero essere esclusi dalle disposizioni di legge sui prodotti, che governano il resto del commercio. Quando questo accade, i produttori fanno di più che appoggiare a parole la questione delle falle di sicurezza, le ripareranno il più in fretta possibile. Ma fino ad allora, l'esposizione totale è l'unico modo che abbiamo per spingere i produttori ad agire responsabilmente.

I motivi che inducono Microsoft a sostenere la segretezza sui bug sono evidenti; è molto più facile sopprimere informazioni sulla sicurezza che risolvere i problemi o realizzare in primis prodotti sicuri. Il flusso costante delle vulnerabilità di Microsoft ha spinto molte persone a mettere in dubbio la sicurezza dei suoi prossimi prodotti. Con analisti come Gartner, che consigliano alle persone di abbandonare Microsoft IIS a causa di tutti i suoi punti deboli, dare ai clienti meno informazioni sulla sicurezza per quanto riguarda i prodotti Microsoft sarebbe ottima cosa per il business.

La segretezza sui bug è una soluzione fruttuosa solo se i produttori di software sono seguaci dei principi di W. Edwards Deming sulla gestione della qualità. Più un bug rimane irrisolto, maggiore è il problema. Siccome il numero di sistemi in Internet è in crescita costante, più a lungo una vulnerabilità rimane irrisolta, maggiore è la finestra di esposizione. Se le aziende credono in questo e agiscono di conseguenza, allora siamo di fronte a una grande discussione sulla segretezza.

Tuttavia la storia dimostra che non accade così. Leggete lo scritto di Culp; lui non dice "ragazzi, se avete un bug mandatemelo e vi assicuro che verrà fissato all'istante." Quel che ha fatto è stato invece contro la pubblicazione di vulnerabilità e chiedere ai ricercatori di tener per sé i dettagli. Altrimenti, ha avvertito, "i produttori non avranno altra scelta che cercare altre strade per proteggere i propri clienti," qualsiasi cosa ciò implichi. Questo è l'atteggiamento che rende l'esposizione totale l'unica strada fruttuosa per ridurre la finestra di esposizione.

Nel suo scritto, Culp paragona la pratica di pubblicare vulnerabilità al gridare "al fuoco!" in un cinema affollato. Ciò che dimentica, però, è che il fuoco c'è davvero, le vulnerabilità esistono a prescindere. Biasimare l'individuo che ha rivelato la vulnerabilità è come mettere in prigione chi ha visto le fiamme per primo. L'esposizione non crea falle di sicurezza, i programmatori le creano, ed esse rimangono finché altri programmatori non le trovano e le eliminano. Ognuno commette errori, sono eventi naturali, nel senso che accadono inevitabilmente. Ma questa non è una scusa per far finta che siano prodotti da forze fuori controllo, e che siano attenuati quando vengono superati.

L'articolo di Scott Culp:

<<http://www.microsoft.com/technet/columns/security/noarch.asp>>

Botta e risposta con Culp:

<<http://news.cnet.com/news/0-1014-201-7819204-0.html>>

Nuovi articoli su Culp:

<<http://www.theregister.co.uk/content/55/22332.html>>



<<http://www.counterpane.com/crypto-gram-0011.html#1>>

Programmare il computer di Satana; perché i computer non sono protetti:

<<http://www.counterpane.com/crypto-gram-9911.html#WhyComputersareInsecure>>

Crittografia a chiave pubblica tramite curve ellittiche:

<<http://www.counterpane.com/crypto-gram-9911.html#EllipticCurvePublic-KeyCryptography>>

Il futuro della frode; tre ragioni per spiegare perché il commercio elettronico è diverso:

<<http://www.counterpane.com/crypto-gram-9811.html#commerce>>

Software di protezione anti-copia; perché non funziona la protezione anti-copia:

<<http://www.counterpane.com/crypto-gram-9811.html#copy>>

\*\* \*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

L'articolo di Scott Culp:

<<http://www.microsoft.com/technet/columns/security/noarch.asp>>

Botta e risposta con Culp:

<<http://news.cnet.com/news/0-1014-201-7819204-0.html>>

Nuovi articoli su Culp:

<<http://www.theregister.co.uk/content/55/22332.html>>

<[http://news.cnet.com/news/0-1003-200-7560391.html?tag=mn\\_hd](http://news.cnet.com/news/0-1003-200-7560391.html?tag=mn_hd)>

<<http://cgi.zdnet.com/slink?153618:8469234>>

Spinte di Microsoft verso la segretezza:

<<http://www.securityfocus.com/news/281>>

<<http://213.40.196.62/media/670.ppt>>

<<http://www.theregister.co.uk/content/4/22614.html>>

<<http://www.theregister.co.uk/content/4/22740.html>>

Il mio articolo originale sulla Finestra di Esposizione:

<<http://www.counterpane.com/crypto-gram-0009.html#1>>

I miei articoli precedenti sull'esposizione totale:

<<http://www.counterpane.com/crypto-gram-0001.html#KeyFindingAttacksandPublicityAttacks>>



Continua la saga SSSCA:

<<http://www.newsforge.com/article.pl?sid=01/10/19/1546246>>

Non si conosce molto a riguardo della legislazione, perché il sen. Hollings si rifiuta di rilasciare ulteriori informazioni e non concede udienze:

<<http://www.newsforge.com/article.pl?sid=01/09/20/2047211>>

Queste persone devono essere fermate, prima che distruggano la sicurezza informatica.

La bozza del SSSCA:

<<http://cryptome.org/ssca.htm>>

Pare che "terrorista" sia un termine molto di moda per accusare qualcuno che non ci piace. Michael Lane Thomas, il maggiore evangelista sviluppatore .Net di Microsoft, ha chiamato coloro che scrivono virus "terroristi industriali". Come ho scritto nel numero precedente, questo è sbagliato e dannoso. I terroristi provocano terrore e non devono essere confusi con criminali qualsiasi. Chi scrive software illecito è fastidioso, provoca danni, costa denaro e distrugge i dati - ma non è un terrorista.

<<http://www.theregister.co.uk/content/56/22423.html>>

<<http://www.zdnet.co.uk/itweek/columns/2001/40/barrett.html>>

Particolarmente odioso è il modo con cui Thomas ha provato ad invocare il patriottismo e l'antiterrorismo, nel tentativo di indurre le persone ad usare prodotti Microsoft. Egli ha detto che se la gente smettesse di utilizzare Microsoft IIS a causa di problemi legati alla sicurezza - come ha recentemente sostenuto Gartner - questo "farebbe solamente il gioco dei terroristi industriali."

Il CERT prevede che per quest'anno gli attacchi informatici saranno in numero doppio rispetto allo scorso anno.

<<http://www.securityfocus.com/news/266>>

<<http://www.zdnet.com/zdnn/stories/news/0,4586,5098301,00.html>>

Ci sono molte cose nuove al NIST. Un rapporto sul secondo workshop sulle modalità operative:

<<http://csrc.nist.gov/encryption/modes/workshop2/index.html>>

Cambiamenti nello standard FIPS 186-2 di Firma Digitale. Fra le altre cose, la nota di cambiamento specifica le dimensioni consigliate delle chiavi e le modifiche al RNG:

<<http://csrc.nist.gov/encryption/tkdigsigs.html>>

Il documento sugli schemi di gestione chiavi riguardante il workshop sopra citato, fissato per l' 1-2 novembre:

<<http://csrc.nist.gov/encryption/kms/workshop2-page.html>>

Un nuovo libro dichiara che un'analista crittografa ha violato parte della macchina codificatrice tedesca Enigma prima della Seconda Guerra Mondiale, ma che i suoi supervisori avevano ignorato le sue teorie.

<<http://www.wired.com/news/women/0,1540,47560,00.html>>

Una nuova versione di Linux sta per essere rilasciata senza informazioni sulla sicurezza, per paura del DMCA. Onestamente non vedo come possa applicarsi il DMCA in questo caso, ma tutto ciò è indicativo del livello di timore all'interno della comunità.

<<http://www.securityfocus.com/news/274>>

<<http://www.securityfocus.com/columnists/35>>

Preoccupazioni di attacchi interni:

<[http://www.computerworld.com/storyba/0,4125,NAV47\\_STO64774,00.html](http://www.computerworld.com/storyba/0,4125,NAV47_STO64774,00.html)>

Un buon articolo sul bisogno di standard per la sicurezza del software:

<[http://www.computerworld.com/storyba/0,4125,NAV47\\_STO64757,00.html](http://www.computerworld.com/storyba/0,4125,NAV47_STO64757,00.html)>

Violare reti wireless:

<[http://news.bbc.co.uk/1/hi/english/sci/tech/newsid\\_1596000/1596033.stm](http://news.bbc.co.uk/1/hi/english/sci/tech/newsid_1596000/1596033.stm)>

È stato violato DRM2, il nuovo software Microsoft per la gestione della sicurezza dei diritti digitali. Uno strumento di hacking sviluppato da qualcuno con lo pseudonimo di Beale Screamer può eliminare la protezione anti-copia dai file audio.

<<http://www.theregister.co.uk/content/55/22354.html>>

<<http://news.cnet.com/news/0-1005-200-7590303.html>>

<<http://cgi.zdnet.com/slink?154661:8469234>>

<<http://cryptome.org/ms-drm.htm>>

Ancora una volta constatiamo che qualsiasi sistema digitale di protezione anti-copia può essere aggirato. Questa violazione non è nemmeno interessante.

Gli hacker di Hong Kong stanno creando un business dalla violazione del sistema di protezione anti-copia di Microsoft:

<<http://www.zdnet.com/zdnn/stories/news/0,4586,2821260,00.html>>

Interessante articolo in quattro parti sulle politiche di sicurezza informatica:

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1193>>

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1473>>

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1487>>

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1497>>

Il furto di identità è in aumento:

<<http://cgi.zdnet.com/slink?154713:8469234>>

Buon articolo sulla guerra dell'informazione:

<<http://www.techreview.com/magazine/nov01/freedmanall.asp>>

Gli IDS e la loro complessità:

<<http://cgi.zdnet.com/slink?154665:8469234>>

Altre novità sul furto della macchina Enigma dal Bletchley Park lo scorso anno. Sono stati ritrovati i rotori:

<[http://news.bbc.co.uk/hi/english/uk/newsid\\_1609000/1609168.stm](http://news.bbc.co.uk/hi/english/uk/newsid_1609000/1609168.stm)>

Articolo in due parti sugli honeypot (i cosiddetti "server esca"), dall'Honeynet Project:

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1492>>

<<http://www.securityfocus.com/cgi-bin/infocus.pl?id=1498>>

Articolo sulla sicurezza e sulle implicazioni riguardanti la privacy dell'iniziativa .NET di Microsoft (scritto da Whitfield Diffie e Susan Landau):

<[http://www.kingpublishing.com/fc/new\\_technology/commentary.htm](http://www.kingpublishing.com/fc/new_technology/commentary.htm)>

Falla di sicurezza in un software che automaticamente aggiorna un antivirus:

<<http://www.zdnet.com/zdnn/stories/news/0,4586,2817368,00.html>>

Alcuni mesi fa avevo scritto un avvertimento per quanto concerne i problemi di sicurezza che Unicode avrebbe portato. Ecco un esempio:

<<http://www.securityfocus.com/bid/3461>>

Network Associates sta cercando di vendere PGP:

<<http://www.wired.com/news/privacy/0,1848,47551,00.html>>

<<http://www.securityfocus.com/news/264>>

Un buon esempio di rischi concreti connessi ai sistemi in rete. Un australiano è stato dichiarato colpevole di essersi introdotto in un sistema di gestione rifiuti informatizzato del Queensland e di aver fatto versare litri di liquame nei parchi e nei fiumi della zona.

<<http://www.theregister.co.uk/content/4/22579.html>>

Altri problemi con Microsoft Passport:

<<http://www.wired.com/news/technology/0,1282,48105,00.html>>

<<http://news.cnet.com/news/0-1003-200-7764433.html>>

<<http://www.washingtonpost.com/wp-dyn/articles/A33656-2001Nov3.html>>

È un rischio enorme il mettere tutte queste informazioni in un unico deposito. L'ufficio Pubbliche Relazioni ha perso completamente di vista il problema. Hanno dichiarato: "In definitiva, ciò che si può dedurre da tutto questo è che non ci sono prove che qualcuno se ne sia mai approfittato." Primo: ci saranno mai delle prove? Secondo: il vero problema sono i rischi futuri, non il danno attuale. Terzo: su quali basi io dovrei continuare a fidarmi delle promesse vacue di Microsoft sulla sicurezza informatica?

Sito Web sulle vulnerabilità di Passport:

<<http://alive.znep.com/~marcs/passport/>>

Il DeCSS è stato dichiarato "pura diceria" da una Corte di Appello della California, andando contro le precedenti disposizioni del tribunale. Buona notizia, quindi.

<<http://www.wired.com/news/print/0,1294,48075,00.html>>

<<http://www.courtinfo.ca.gov/courts/courtsofappeal/6thDistrict/>>

<<http://slashdot.org/yro/01/11/01/1953236.shtml>>

<<http://www.theregister.co.uk/content/55/22613.html>>

Un cellulare GSM con crittografia end-to-end:

<<http://www.pcworld.com/news/article/0,aid,51368,00.asp>>

Articolo sulle problematiche della "sicurezza senza informazioni". Viene sottolineato il fatto che Microsoft probabilmente batterà il suo record di patch rilasciate quest'anno, più di 100 (cioè due alla settimana).

<<http://www.vnunet.com/Analysis/1126488>>

Ad alcune ore dal rilascio di Windows XP, alcuni pirati hanno violato gli schemi di protezione anti-copia e hanno iniziato a distribuire copie piratate.

<<http://www.newsbytes.com/news/01/171651.html>>

Il panico di Microsoft in questo caso ha ragion d'essere. Vi sono certamente rischi di sicurezza nell'utilizzo di software pirata.

Altri dati sulla sicurezza; un'indagine afferma che un server IIS su nove può essere conquistato dagli hacker. Ci si meraviglia ancora che Gartner stia consigliando alla gente di cambiare server?

<<http://www.infoworld.com/articles/hn/xml/01/11/02/011102hnsurvey.xml>>

Un'altra indagine sostiene come i due terzi delle reti wireless londinesi siano completamente indifese:

<[http://news.bbc.co.uk/hi/english/sci/tech/newsid\\_1639000/1639661.stm](http://news.bbc.co.uk/hi/english/sci/tech/newsid_1639000/1639661.stm)>

Gli strumenti di hacking stanno diventando più agguerriti:

<<http://www.computing.vnunet.com/News/1126643>>



uno di quei risibili prodotti "Air-Gap". GOVNET deve utilizzare dei propri router, dei server e dei client proprietari. Se un utente GOVNET vuole accedere ad Internet deve avere due computer sulla propria scrivania. Può utilizzare gli stessi programmi su entrambe le macchine, ma devono essere due copie diverse. Non può condividere i file, neanche via floppy.

Venir meno a una qualsiasi di queste regole significa compromettere la sicurezza. Si provi a passare un floppy di Microsoft Word da una rete all'altra, e c'è il rischio di infezione tramite virus macro. Si colleghi un computer ad entrambe le reti, e si rischia l'introduzione di ogni genere di software degli hacker. Si aggiungano punti di accesso dial-up pubblici, e ci saranno dei tentativi di intrusione dall'esterno.

GOVNET non è un'idea nuova. Esistono già delle reti separate all'interno del Governo statunitense - INTELINK, SIPRNET, NIPRNET, ecc. - alcune di queste sono reti segrete. Le reti segrete sono completamente criptate e tutti i punti d'accesso sono situati in stanze ed edifici di massima sicurezza. Sono assai più sicure di Internet, tuttavia al virus Melissa occorsero 24 ore per penetrare da Internet all'interno di una di queste reti. Anche il virus LoveLetter infettò parecchi di questi computer.

Posso immaginare che cosa sia accaduto. Qualche dirigente ha controllato la propria email su Internet. Poi ha collegato il suo portatile ad una di queste reti segrete, sicure e separate, e i virus sono penetrati.

Ma anche un'idea di rete simile è assai meglio di ciò che abbiamo oggi. Una GOVNET progettata da zero può avere molte funzioni di sicurezza in più. Può essere realizzata un'autenticazione più efficace (visto che i prodotti commerciali la permettono). Tutti i link possono essere criptati. L'anonimato può venire bandito. Ci può essere maggiore responsabilità. Può essere approvato un elenco di software lecito. GOVNET potrebbe non essere in grado di impedire attacchi interni, ma potrebbe sicuramente rendere la cosa molto più difficile.

D'altro canto, separare fisicamente una rete da Internet la rende anche molto meno utile. L'utilità è il criterio che ha spinto le aziende a connettere le proprie reti ad Internet fin da subito. Per molti aspetti questo risulterebbe essere un passo indietro. Internet ha avuto questo nome perché era una rete di reti. Ai vecchi tempi c'erano Arpanet, Milnet, BITnet, Usenet, JANET, e una serie di altre reti disgiunte. Connetterle fra loro le ha rese tutte più utili.

Dal momento che GOVNET (e le altre) si sconnettono da Internet, diventano di conseguenza meno utili. Reti come INTELINK hanno dei compiti ben specifici: ecco perché funzionano. GOVNET non ne ha, e questa è la sua debolezza maggiore. Gli utenti avranno la necessità di accedere ad Internet, e la tentazione di collegarsi ad Internet attraverso una specie di firewall sarà sempre in agguato. Ecco che la separazione va a farsi benedire. Sfortunatamente, la sicurezza di una cosa come GOVNET è destinata ad essere inversamente proporzionale alla sua utilità.

Comunicato stampa:

<<http://w3.gsa.gov/web/x/publicaffairs.nsf/dea168abbe828fe9852565c600519794/1c10e9ac670553b885256ae100668beb?OpenDocument>>

Novità e commenti:

<[http://news.bbc.co.uk/low/english/sci/tech/newsid\\_1601000/1601823.stm](http://news.bbc.co.uk/low/english/sci/tech/newsid_1601000/1601823.stm)>

<<http://www.theregister.co.uk/content/archive/22156.html>>

<<http://www.zdnet.com/zdnn/stories/news/0,4586,5098134,00.html>>

<<http://www.zdnet.com/zdnn/stories/news/0,4586,5098169,00.html>>



Descrizione della vulnerabilità:

<<http://cert.uni-stuttgart.de/archive/bugtraq/2001/09/msg00158.html>>

\*\*\* \*\*

Microsoft a riguardo di Windows XP

eWeek ha riportato un'intervista con Jim Allchin, vicepresidente del Microsoft Platform Group. Vorrei riportarne un paio di stralci e commentarli.

"Windows XP è incredibilmente più sicuro di Windows 2000 o di una qualsiasi versione precedente. L'overflow del buffer è stato uno degli attacchi più frequenti in Internet. Abbiamo fatto passare tutto il codice ed abbiamo trovato, grazie a strumenti automatizzati, punti in cui era possibile creare un overflow del buffer, e questi punti sono stati rimossi da Windows XP."

"Abbiamo inoltre eliminato per default tutta una serie di cose, in modo tale che ora gli utenti vengono configurati più semplicemente, rendendoli così meno vulnerabili. Abbiamo anche collegato ad Internet una macchina Windows XP senza protezioni, invitando le persone a tentare di intaccarla. Finora non ci sono state intrusioni o problemi."

Mi piace conservare queste citazioni. Ogni volta che Microsoft rilascia un sistema operativo, dichiara che questo è incredibilmente più sicuro del precedente. Lo ha affermato sia per Windows NT che per Windows 2000. Ogni volta non si è dimostrato vero. Torneremo su questi stralci fra un annetto circa.

"Abbiamo testato tutte le riparazioni ai buchi di sicurezza. Con tutte quelle che abbiamo pubblicato da dieci anni a questa parte, ci stiamo muovendo verso una regressione per ognuna di esse; le informazioni vengono diffuse sul sito Microsoft prima di essere pubblicate ufficialmente, tutto viene verificato qui in produzione, e siamo molto sicuri per quanto concerne la qualità del prodotto... le nostre patch non includono nuove funzionalità, ma sono progettate specificatamente per eliminare una possibile intrusione."

Non abbiamo neanche dovuto aspettare un anno per questa; nello stesso giorno dell'intervista, Microsoft ha dovuto produrre immediatamente una patch di sicurezza perché le reti degli utenti erano praticamente senza difese dopo l'installazione. Oops. Questa sarebbe la "regressione", questo sarebbe il testare le patch sulla rete di Microsoft.

Un'altra patch di Microsoft, quella che serve a riparare l'odiosa vulnerabilità descritta in MS01-50, è difficile da installare, in parte a causa della "decisione di Microsoft di inserire una serie di patch che nulla hanno a che fare con la sicurezza, come la correzione di un errore nell'ordinamento alfabetico degli elenchi in lingua Ceca in Excel" (fonte: Business Week).

Ho sostenuto per molto tempo che Microsoft tratta i problemi inerenti alla sicurezza più che altro come problemi di Pubbliche Relazioni, e tutto questo non fa altro che dimostrarlo. Microsoft dichiara spudoratamente il falso alla stampa quando parla di sicurezza, come possono ampiamente dimostrare i suoi comportamenti ogni volta.

L'intervista ad Allchin in eWeek:

<<http://www.eweek.com/article/0,3658,s%253D701%2526a%253D16895,00.asp>>

La patch difettosa di Microsoft:

<[http://www.computerworld.com/storyba/0,4125,NAV47\\_STO64947,00.html](http://www.computerworld.com/storyba/0,4125,NAV47_STO64947,00.html)>



Quali gli effetti nel commercio? Che ne sarebbe delle multinazionali americane con uffici in Europa? Sarebbe possibile o anche obbligatorio? Vi sarebbe forse una legge per imporre ad un'azienda americana di scambiare solamente file conformi allo standard SSSCA? Le implicazioni sarebbero su vasta scala: un'azienda non potrebbe più acquistare in un libero mercato, ma in un mercato conformato al SSSCA, presumibilmente con un aumento dei costi. Ecco quindi il profitto.

Da un certo punto di vista, l'America arriverebbe al suicidio economico. Da un altro, potrebbe rivelarsi solo un enorme inconveniente per chi acquista computer dell'ultima generazione. Magari i vecchi computer pre-SSSCA diventerebbero così popolari che la gente finirebbe col comprarli al mercatino dell'usato!

In realtà, ritengo che la storia abbia dimostrato come ogni schema di protezione anti-copia possa venire sconfitto. Nel peggiore dei casi, questo si rivelerà uno svantaggio per i cittadini e le aziende americane. Nel migliore, invece, la gente si renderà conto di tutto questo e il disegno di legge non verrà mai varato.

Da: Anonimo

Oggetto: Protezione anti-copia di CD e DVD

- > Ho ribadito più volte che l'industria dell'intrattenimento non vuole che la
- > gente abbia dei computer. I computer danno agli utenti troppe possibilità,
- > troppa versatilità, troppa libertà. L'industria dell'intrattenimento invece
- > vuole che la gente si sieda in poltrona e consumi, e sta cercando di
- > trasformare il computer in una Piattaforma di Intrattenimento Interattivo
- > attraverso Internet, sulla scia di quello che già sono il televisore e il
- > videoregistratore.

CENTRO!

Lavoro per un'azienda che sviluppa hardware per CD e DVD. Sono continuamente sorpreso dalle idee che sono emerse qui, idee che riguardano il trasformare i CD e i DVD in sistemi pay-per-play.

Il mio datore di lavoro tiene queste idee molto in considerazione perché possono sfociare in una causa da parte dell'industria dell'intrattenimento, intentata sulla base di una "fuga di notizie", cioè di rottura del copyright per conto terzi.

Noi non piratiamo la musica o i film, ma i pirati utilizzano il nostro hardware su un PC, e quindi potremmo essere ugualmente perseguibili in qualche modo. Questo è il timore, comunque, e quindi ogni tanto qui sul lavoro viene proposto un nuovo formato: dischi audio tipo DIVX, videodischi pay-per-play, ecc.

Se lei è intenzionato a vedere la versione che l'industria dell'intrattenimento sogna per CD e DVD, come dovrebbero essere e come dovrebbero venire usati, vada al sito <<http://www.dataplay.com>>

Praticamente l' "uso corretto", nel comune significato del termine, è completamente eluso dall'hardware. Bisogna pagare per usufruire dei dati, anche se questi sono in un disco nelle proprie mani, regolarmente pagato coi propri soldi.

Da: Martin Rex <[martin.rex@sap-ag.de](mailto:martin.rex@sap-ag.de)>

Oggetto: Responsabilità e software

Il discorso di Bruce sulla responsabilità riguardante il software è assolutamente corretto, così come il paragone coi pneumatici Firestone.

Code Red sfrutta le specifiche **interne** di IIS, dato che effettua richieste che sono valide sia per le specifiche HTTP/1.0 sia 1.1. Chiaramente è un problema del server web se esso non è in grado di far fronte correttamente ad una richiesta valida.

Allo stesso modo con Outlook, dove i settaggi (della zona di sicurezza) dichiarano di offrire protezione da file eseguibili o da altri allegati non sicuri. Invece i sistemi implementati da Microsoft non hanno mai funzionato seriamente. Nimda è composto solo da caratteri, è Outlook che lo fa diventare un virus. Il mio client di posta elettronica, basato su Unix e totalmente a caratteri mi farebbe vedere Nimda per quello che realmente è: caratteri.

Se un pneumatico Firestone scoppia dopo uno specifico schema di sterzate "sinistra, destra, sinistra, sinistra, sinistra, destra, sinistra", il costruttore non può cavarsela dicendo che questo tipo di schema d'uso è improprio e che quindi non ne è responsabile.

Se un costruttore dichiara esplicitamente la propria falciatrice sicura per i bambini di qualsiasi età, allora può essere chiamato in causa sicuramente nel caso i bambini, giocandoci, si facciano del male.

Microsoft sta vendendo software che viene dichiarato "sicuro" per Internet, anche se in realtà non è affatto sicuro, qualsiasi sia il significato pratico del termine "sicuro".

Sicuro per Internet significa che il loro server IIS non solo gestisca correttamente gli URL approvati da Microsoft ma anche tutti i protocolli HTTP/1.0 e 1.1 validi, compresi i pattern usati da Code Red e da Nimda.

Stessa cosa per Outlook; i programmi di e-mail **devono** essere sicuri per ogni uso permesso dalle specifiche dei protocolli su cui si basano, non solo su quelli testati da Microsoft.

Da: Edward Welbourne <[eddy@vortigen.demon.co.uk](mailto:eddy@vortigen.demon.co.uk)>

Oggetto: Re: Responsabilità e Software

Nell'ultimo numero di Crypto-Gram, Buck Hodges <[ewhodges@yahoo.com](mailto:ewhodges@yahoo.com)> ha scritto:

- > L'abuso dei difetti è ciò che differenzia tutto questo dalla responsabilità
- > legata ai prodotti tradizionali. Dovremmo invece perseguire i criminali che
- > causano disastri e caos intenzionalmente.

Quindi, a occhio e croce, Buck sostiene che worm, virus, ecc. dovrebbero essere paragonati a qualcuno che svuota una scatola di puntine in mezzo alla strada. Il costruttore dei pneumatici non è responsabile per ogni strage che ne deriva, il vandalo sì.

Tutto ciò ha abbastanza senso, ma se un fabbro cambia le serrature della porta del vostro appartamento con alcune tremendamente semplici da scassinare, ci dovrebbe essere, e forse ci sarebbe davvero, un caso di responsabilità connessa al prodotto. Almeno nel caso in cui si subisca un furto immediatamente dopo il cambio di serratura senza che il fabbro vi abbia avvertito del rischio. Il ladro si è servito di difetti e debolezze per commettere reato, ma un certo grado di responsabilità è imputabile anche al fabbro, per non aver reso la vita difficile al ladro.

- > Molti dei difetti [...] forzati da worm o da virus non avrebbero alcun
- > effetto sul prodotto se il prodotto fosse usato correttamente

Ecco il punto cruciale: lascia la questione tutt'altro che risolta.

Una grande differenza fra il software e i pneumatici sta nel considerare l'"uso corretto del prodotto".

Con le automobili è semplice: se si sta guidando su una strada pubblica in buone condizioni, ad una velocità sicura considerate le condizioni della strada in assenza di ghiaccio, macchie d'olio, puntine, lame, vetri o altre situazioni anomale, si sta usando il prodotto correttamente. Vi sono alcune altre situazioni in cui si può affermare che si sta usando il prodotto correttamente, ma il caso illustrato sopra copre la maggior parte degli usi a cui si possono sottoporre i pneumatici di un'auto. Le eccezioni riguardano una certa variazione dei limiti sopra elencati.

Con le serrature le cose si fanno meno nette, anche se mi sento di dire che c'è una maggiore casistica legale rispetto al software; direi che le falle di sicurezza nel software dovrebbero essere considerate in maniera più affine alle serrature che non ai pneumatici. Ma la situazione, nel caso del software, è più complicata.

Ad esempio...di recente il mio server ha ricevuto una richiesta (il 5/8/2001):

```
GET /default.ida?...%u00=a HTTP/1.0
```

dove "... " comprendeva una serie di 224 "N", seguiti da una sequenza di 22 token nel formato %uxxxx, dove ogni x sostituisce una cifra esadecimale, 0-9 o a-f. Infatti i 22 token in questione erano: %u9090 %u6858 %ucbd3 %u7801 %u9090 %u6858 %ucbd3 %u7801 %u9090 %u6858 %ucbd3 %u7801 %u9090 %u9090 %u8190 %u00c3 %u0003 %u8b00 %u531b %u53ff %u0078 %u0000.

Ora, per quanto ne so, questa è una richiesta HTTP correttamente formulata. Non molto elegante, certo, ma dando un'occhiata ad alcuni link inseriti nell'ultimo numero di Crypto-Gram si può notare che molti di essi finiscono col diventare una serie di puntatori e scritte incomprensibili dopo l'iniziale formato "protocollo://sito.dominio", a volte con un ":port" e forse anche parte del percorso. Non è così strano, dopotutto. Sebbene, lo ammetto, non è affatto innocente.

Ho un web server installato sul mio computer e lo uso correttamente. Ovvero, il mio computer è configurato per delegare al server la gestione di richieste in entrata sulla porta 80, la porta di default per HTTP. Dato che quella vista sopra è una richiesta HTTP legittima, indirizzata a quella porta, posso affermare che tutte le parti coinvolte (compreso il richiedente) stavano usando il prodotto "correttamente".

Sono lieto di poter dire che non faccio uso di IIS, e quindi il mio server web ha doverosamente tracciato la richiesta, unitamente a data, ora, indirizzo IP originario (ho mandato al richiedente un'e-mail educata, suggerendo loro di verificare la presenza di un virus o di un worm), il codice di risposta HTTP (404) e il numero di byte inviati in quella risposta. Il mio web server si è comportato correttamente.

Però un'installazione di default di IIS che avesse ricevuto la stessa richiesta (che, a quanto pare, era la mossa di apertura di Code Red) sarebbe stato dirottato verso un programma che il proprietario dell'installazione non avrebbe mai permesso di far girare su quel computer.

In queste circostanze non vedo come IIS sia stato usato "scorrettamente". Ammetto che qualche altro software - qualcosa richiamato come risultato di un accesso a /default.ida su una macchina IIS - non fosse affatto usato correttamente. Comunque, dato che il processo di installazione di Microsoft per IIS stabilisce automaticamente che IIS deleghi quell'altro software in casi analoghi, senza far presente all'utente il rischio di tutto questo, chi non sta utilizzando correttamente il prodotto in questo caso è Microsoft stessa (attraverso il proprio

software, almeno). A me sembra che tutto questo sia molto simile al caso dei pneumatici Firestone.

Non nego che ci sia spazio per ulteriori argomentazioni, dico solo che le acque sono molto più torbide rispetto al caso dei pneumatici. Sospetto che simili argomentazioni possano essere applicate ad Outlook e a molti altri programmi preferiti da chi realizza worm e virus.

D'altro canto sono d'accordo con Buck per quanto riguarda i pericoli dell'estensione del concetto di responsabilità sul prodotto al software. È una cosa talmente complessa che non si sarebbe in grado nemmeno di rilasciare alcun software per paura di cause che coinvolgano la responsabilità sul prodotto. Credo che sia questo il motivo per cui tutti possiedono una licenza che declina ogni responsabilità - una pratica per la quale porto maggiore pazienza quando il codice sorgente del prodotto è liberamente disponibile, rispetto a quando viene tenuto segreto - per evitare inconvenienti del genere.

- > A un sacco di avvocati piacerebbe poter intentare cause di enorme portata ai
- > danni di Microsoft, IBM, Adobe, Apple, ed altri (anche Red Hat) per le
- > responsabilità connesse ai danni causati da falle di sicurezza forzate da
- > worm, da virus, o da altri programmi altrettanto pericolosi. Farebbero una
- > fortuna, e noi consumatori pagheremmo i vari procedimenti legali grazie a un
- > software più costoso.

Come conseguenza di ogni attacco di un worm, leggiamo in vari comunicati stampa quanti milioni di dollari sono andati perduti per colpa di quell'attacco. Sono portato a sospettare che dichiarazioni del genere siano un tantino gonfiate, ma in ogni caso, siamo senza dubbio di fronte a delle perdite consistenti. Un buon avvocato sarebbe in grado di farle credere incredibilmente grandi, almeno quanto si vocifera.

Molte delle perdite vengono sostenute dalle aziende. La perdita è grave perché si fa uso dei prodotti Microsoft, che hanno una lunga storia di fallimenti in ambito di sicurezza. Esistono soluzioni software meno rischiose, specialmente per quanto concerne i contendenti di IIS, e quindi molte di quelle perdite sarebbero evitabili. I dirigenti e gli impiegati delle aziende devono preoccuparsi, nei confronti degli azionisti, di evitare rischi costosi, ove possibile. Il fatto che un reato è stato commesso contro l'azienda non li giustifica, né più né meno che in un caso di furto reso possibile dalla trascuratezza di tutto il personale nel chiudere porte, finestre, casseforti, ecc. una volta usciti dagli uffici.

Così penso che sia arrivato il momento in cui certi azionisti denunciino certe aziende, comitati direttivi o manager dello staff tecnologico, per mancanza di "correttezza dovuta" che ha causato perdite effettive di capitale (come riportato dalla stampa). Oppure alcuni legali potrebbero intentare una causa per conto di alcuni azionisti.

Dato che Microsoft utilizza molto del suo stesso software e talvolta accusa perdite di capitali dovute a worm, virus, ecc., sarebbe particolarmente amaro e ironico vedere i suoi azionisti denunciare i dirigenti per irresponsabilità nell'uso dei loro stessi prodotti.

\*\* \*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\* \*\*\*\*\*

CRYPTO-GRAM è una newsletter mensile gratuita che offre riassunti, analisi, approfondimenti, e commenti sulla sicurezza informatica e sulla crittografia.

La versione italiana è curata da Communication Valley <http://www.communicationvalley.it>; per iscriversi o cancellarsi andare all'indirizzo <http://www.cryptogram.it>. I numeri arretrati sono disponibili all'indirizzo <http://www.cryptogram.it>. Per informazioni [crypto-gram@communicationvalley.it](mailto:crypto-gram@communicationvalley.it).

Inoltrate liberamente CRYPTO-GRAM a colleghi ed amici che possano trovare la rivista interessante. Viene concessa l'autorizzazione a ristampare CRYPTO-GRAM, a condizione che sia stampata integralmente.

CRYPTO-GRAM è realizzata da Bruce Schneier. Schneier è il fondatore e CTO di Counterpane Internet Security, Inc., autore di "Secrets and Lies" e di "Applied Cryptography" e inventore degli algoritmi Blowfish, Twofish e Yarrow. Ha prestato la sua collaborazione al comitato dell'Associazione Internazionale di Ricerca Crittografica (International Association for Cryptologic Research), ed è membro del comitato consultivo dell'Electronic Privacy Information Center (EPIC). Scrive spesso e tiene conferenze in merito alla sicurezza informatica e alla crittografia.

Counterpane Internet Security, Inc. è leader mondiale nel monitoraggio guidato della sicurezza informatica. Gli analisti esperti in sicurezza di Counterpane proteggono reti per aziende inserite nella Fortune 2000 a livello mondiale.

<<http://www.counterpane.com/>>

Copyright © 2001 by Counterpane Internet Security, Inc.