FALLIMENTI PROMETTENTI
hackmeeting 2007 - Pisa

idee disparate, abbandonate durante la loro crescita, discusse al fine di chiaccherare della loro utilità e di tecnologie analoghe.

filo logico:

- obiettivo
- contesto/idea/implementazione
- dettagli/considerazioni

vecna - www.s0ftpj.org www.winstonsmith.info www.delirandom.net

edera - obiettivo

- creare mailing list anonime
- sfruttare un'aggregazione gia' esistente (garantisce una certa diffusione, consente una correlazione inferiore)

edera – formulazione

- le mailing list accettano una mail in relazione al mittente
- SMTP non puo' verificare il mittente

edera - idea

- utenti non iscritti possono inviare su una mailing list
- l'accesso e' autenticato ma non identificato
- ogni e-mail e' firmata dal sistema edera
- per ogni “from” si genera uno pseudonimo

edera - implementazione

- anonimato delegato all'infrastruttura (tor/remailer/formmail :)
- si deve avere una lista degli iscritti aggiornata
- una lista di pseudonimi lunga
- una [tag] grazie alla quale poter filtrare

links (utili ?)

http://www.gnu.org/software/anubis/

unid - obiettivo

- rendere piu' difficile l'open source intelligence (senza soffrire degli svantaggi di un anonimato compulsivo :)
- consentire il diritto all'oblio dai motori di ricerca

unid - contesto

- chi scrive su sistemi pubblici lascia tracce riconducibili a lui tramite Open Source Intelligence
- questa caratteristica e' necessaria alla comunicazione, ma puo' rivoltarsi contro di noi

unid - contesto

- l'OSI viene fatta su quei parametri correlati all'utente (nome/nick/mail)
- questi parametri, se resi one-time, mantengono le feature e tolgono la linkability
- puo' essere temporale

unid – idea – quello che fa l'utente

- genera un codice univoco, lo firma e lo invia al server
- se viene accettato, lo usa come pseudonimo/email/alias/nick/firma/...
- con la stessa chiave puo' revocare l'unid

unid – idea – quello che fa il servizio

- al posto del proprio identificativo viene messo un codice univoco che viene memorizzato da un server, firmato con la nostra chiave pubblicata su keyserver
- server unid memorizza unid univoco + firma

unid – idea – quello che fa il lettore

- trovando un codice unid lo “risolve” tramite il servizio di risoluzione. il server restituisce la richiesta firmata dell'autore (dal quale si deriva l'identita')
- cerca di automatizzare a colpi di javascript

unid – effetto del “layer aggiuntivo”

- mancanza di un pattern fisso
- autenticazione
- ripudiabilita' programmata o on-demand
- possibile aggiunta di dati extra

unid – come puo' apparire ?

- ad ogni email su lista, il client mette al posto di “From:” qualcosa come From: unid::N408NV938938@unidsrv.tw
- dove non e' automatizzabile, lo si genera via web/script

unid – quali componenti ?

- server che riceve, convalida, memorizza, verifica unid + firma (check su keyserver)
- lo stesso che puo' ricevere le revoche
- una pagina di interrogazione “unid::3F4F30N3G39”=“vecna <vecna@x.y> 2/3/02 [commenti]”

unid – quanto deve essere sicuro ?

- si affida ai keyserver, funziona solo verificando firme
- il server accetta o rifiuta un unid proposto
- le comunicazioni per la “risoluzione” usano HTTPS

links :(

mod_blob - obiettivo

- consentire l'accesso ad una pagina web solo ad un client e non ad uno strumento. come discriminante viene usata la presenza di javascript, assente in sniffer, crawler, script

mod_blob - contesto

- esistono sw (php/js) che convertono una pagina HTML in un blocco cifrato
- i webserver possono dare un “blob” js al posto dell'HTML, il browser l'esegue e ti decifra la pagina.

mod_blob - idea

- fare un modulo per apache perche' lo faccia in automatico ?
- decidere se ci si vuol difendere da: un attacco mirato (password) un attacco di massa (captcha) o da un SE (decifrazione automatica)

mod_blob – impatto

- il web ha aumentato cosi' tanto complessita' che non so prevedere dei problemi (!?)
- nascita di implementazioni automatiche in php (forse e' meglio, perche' si ha accesso piu' spesso a quel layer ?)

mod_blob – dubbi implementativi

- apparentemente semplice su content-type: text/html, ma rinunciamo agli altri ?
- difficolta' con ajax
- solo le pagine ricevute sono cifrate o anche le GET/POST inviate ?

links (utili ?)

http://messagevault.org/
http://jgae.de/sdaeng.htm
http://pajhome.org.uk/crypt/sda/index.html

metacrypto - obiettivo

- proteggersi da attacchi di crittanalisi non ancora noti
- rimanere coerenti con i capisaldi della sicurezza

metacrypto - contesto

- tutta la mia vita dipende da RSA
- anche la vita di alcuni amici dipende da RSA
- pensa a freebsd 5.0
- pensa alla ricerca spicciola
- la crittografia e' DELICATA

metacrypto - idea

- utilizzare solo algoritmi standard
- MA, non dare “elementi fissi”
- sia attacchi che software si basano su assunzioni
- si realizza un protocollo che gestisca la rotazione di questi elementi

metacrypto - idea

- avere un protocollo di rotazione chiavi/algoritmi non e' “security by obscurity”
- idem cambiare la chiave a tempo random/preudorandom
- idem cambiare l'algoritmo a tempo random/pseudorandom

metacrypto - idea

- non mi interessa mettere “un bit in piu'” il bit in piu' protegge dal bruteforce, dal bruteforce mi proteggono gia' 256 bit, ma dalla matematica chi mi protegge ?

metacrypto – idea (simmetrica)

- lunghezza della chiave non nota
- numero e grandezza delle s-box non noto
- aggiornamento della chiave
- sacrificio del calcolo
- algoritmi a sorpresa
- utilizzo di checksum (?)

metacrypto – idea (asimmetrica)

- sequenza di algoritmi differenti
- chiavi di lunghezze differenti
- sacrifico del calcolo
- utilizzo di un tempo fisso (forza dipendente dalla CPU)

metacrypto - considerazioni

- kazaa
- storicamente perche' no
- ma praticamente, perche' no ?

links, molto generici

http://freehaven.net/anonbib/date.html

proxynhood - obiettivo

- rendere la correlazione e le analisi di MASSA parzialmente inutili
- necessaria la collaborazione di una percentuale di massa
- visto che al problema e' insensibile, trovare una scusa irresistibile ;P

proxynhood – contesto (reale)

- le correlazioni aumentano (sameorigin, cookie cross domain)
- le discriminanti sono cookie/variabili del browser esportabili via js
- il gioco anti-esselunga
- TrackMeNot pseudo-fallisce

proxynhood – contesto (apparente)

- i banner piu' clickati fanno guadagnare
- i primi in classifica sono i piu' clickati
- le click fraud vengono facilmente individuate (in quanto “picchi statistici”)

proxynhood – idea (reale)

- cosa confonde l'analisi ? delle coperture non casuali (quindi auto-analisi dell'impronta) o delle fusioni di impronte
- un proxy che centralizza le ricerche e ruota cookie, simula ricerche...

proxynhood – idea (apparente)

- se i primi sono i ricchi
- se i ricchi si arricchiscono
- se automaticamente io clicko a quelli in terza, quarta, quinta pagina
- rubo a google per dare ai poveri!
- la cosa va EQUILIBRATA BENE!

proxynhood - idea

- proxare diversi utenti
- proxare e rotare diversi cookie
- simulare la ricerca correlata
- spingersi su pagine di ricerca oltre la prima
- random (1/30 ?) click su ADS dei link in ultima pagina

proxynhood - idea

- non si devono creare anomalie
- i click devono essere rari e distribuiti (previo ban)
- le ricerche devono essere coerenti (magari piu' aperture di pagine e meno ricerche)
- (si puo' fare meta-SE) ?

proxynhood - impatto

- puo' davvero influenzare ADS ? non importa neppure :)
- si ottiene la fusione delle impronte
- per combattere un'analisi di massa serve una percentuale di massa

proxynhood – mah ...

- E' google il problema o dove i dati transitano ?
- Avrebbe senso fare un google sniffer ? (semplice sniffer HTTP + analisi dei cookie)
- = E' piu' efficace dar soluzioni o dar problemi ?

links (utili ?)

http://www.readwriteweb.com/archives/top_100_alternative_search_engines.php

http://www.google-watch.org/

FINE PRIMO TEMPO

- perplessita'
- turbamenti
- commenti
- suggerimenti

remind: proxynhood, metacrypto, mod_blob, unid, edera

PARANOIA MODE ON

- secondo l'analisi dei modelli di minaccia, quello che segue non e' sensato.
- *ma e' phun :)*

plausible deniability

- E' la possibilita' di negare la presenza di dati all'interno di un contenitore
- Trucrypt
- c2c
- implicita nella steganografia

l'efficacia della plausible deniability

- ci sara' da qualche parte (disco, chiavetta, cache) un sw che consente la plausible deniability
- l'uomo e' l'anello debole (paranoia, tortura, interrogatori)

steganografia su filesystem

- sfs: dati protetti in R/W, resto del disco in read only
- steganografia classica su media
- FUSE
- modularita' di FUSE

ilma – divagazione

- IL weird language “brainfuck”
- la dervizione delle istruzioni dal web (antiforense)
- la possibilita' di farle cambiare nel tempo (codice mutante :)

blastersteg -> noisesteg (contesto)

- il tuo traffico, se analizzato, viene catalogato
- la catalogazione viene fatta per analisi grossolana e solo dopo per analisi approfondita
- in modo automatico noi diamo molti input
- e riceviamo ancora piu' noise

obiettivo finale

- non ci sono dati nascosti (se resisto alla steganalisi)
- non ci sono software di steganografia (lo ricevo ogni volta)
- non ho richiesto nulla di esplicito o discriminate

implementazione rete / open issue

- il codice e' tanto. usare lo spam ? usare traffico p2p ? skype ? joost ?
- utilizzo del timing
- ricezione del codice o di codice di sblocco ?

implementazione FUSE / open issue

- c'e' da “mappare” lo spazio disponibile sui contenitori
- integrare i software di stegano o lasciarli separati ?
- c'e' da trovare un ordine ?
- la quantita' di disco puo' aumentare ?

steganografia over spam - ?

- quanto spam e' leggibile ?
- lo spam supera il problema della rete “piccolo mondo”
- puo' non essere monodirezionale
- (come edera, lo si implementa con anubis ?)

Plausible Deniability ToolKit

http://www.nmrc.org/pub/pdtk/

http://www.stefaniacastoldi.it

destination spoofing - obiettivo

- essere anonimi non prima di un attacco, ma durante l'acquisizione dei dati
- prevedere le analisi forensi, IDS e sniffer sulle reti d'appoggio
- sfruttare qualcosa non percepito come -delicato-

destination spoofing - contesto

- per “spoofing” si intende anonimizzare il mittente
- il problema e' che il ricevente non sa' chi e' il vero mittente
- gli attacchi vecchio stile non sono piu' applicabili

destination spoofing - contesto

- non e' anonimizzare un attacco, il problema reale
- e' anonimizzare la fuga senza la perdita d'efficenza
- ogni grosso trasferimento puo' essere visto come anomalia, e la destinazione subire analisi

destination spoofing - idea

- inviare i dati a una destinazione falsa (di sola deception per l'analista)
- intercettare il traffico sul tragitto
- rigirarlo verso di noi
- quanti router aperti esistono su internet :P ?

destination spoofing – open issue

- openvpn (tcp over udp/TLS)
- MITM + alias interfaccia + DNAT (ma evitiamo le macchine a questo punto)
- necessario conoscere i router nel loro dettaglio

links ?

apmislay: aka, “anonimato senza rimbalzi” - obiettivo

- l'anonimato convenzionale per il file sharing consuma risorse.
- trovare un metodo meno oneroso, possibilmente senza rimbarlzi
- va protetto il mittente ?
- l'ip spoofing (non muore mai)

apmislay - idea

- il mittente manda i dati da una sorgente anonima
- il ricevente deve mandare solo degli ACKs, una quantita' di dati minimi
- il ricevente cerca di indovinare il mittente
- non su 32 bit, ma su 4,5,...

apmislay - implementazione

- e' una connect semplice + un po' di colpi di iptables
- e' gia' implementato come libreria
- i reset vanno ignorati ed usati per apparire come gli altri

link!

http://www.s0ftpj.org/it/tools.html

http://www.stefaniacastoldi.it

FINE SECONDO TEMPO

- perplessita'
- turbamenti
- commenti
- suggerimenti

remind: apmislay, destination spoofing, steganografie estreme

vecna@(delirandom.net|s0ftpj.org|winstonsmith.info)