

Security-By-Contract: come usare software scaricato da internet sul proprio telefono senza pentirsene...

Nicola Dragoni

dragoni@disi.unitn.it

Fabio Massacci

Fabio.Massacci@unitn.it

www.massacci.org

Dipartimento di Ingegneria e Scienza dell'Informazione
Università degli studi di Trento

Abstract: In questo articolo viene presentato il paradigma del Security-By-Contract (SxC), un approccio per migliorare la sicurezza nei dispositivi mobili sviluppato nel contesto del progetto europeo S3MS. L'intuizione di base è che un'applicazione software viene scaricata insieme ad un "contratto" che descrive tutte le interazioni rilevanti, in termini di sicurezza, che l'applicazione avrà col dispositivo mobile sul quale verrà eseguita. Il contratto dovrà essere quindi accettato dal dispositivo in funzione di una policy di sicurezza definita dall'utente o dall'operatore della rete. Tale paradigma non sostituirà gli attuali sistemi di sicurezza ma li migliorerà, fornendo un meccanismo di sicurezza flessibile, semplice e scalabile.

1. Il Problema della Sicurezza nei Dispositivi Mobili

Negli ultimi anni i dispositivi mobili sono diventati ormai oggetti di uso comune. Si pensi ad esempio ai telefoni cellulari, che hanno raggiunto una potenza di calcolo in grado di eseguire applicazioni software sempre più complesse (giochi, strumenti per l'ufficio quali agenda, calendario, email, ecc...). Non a caso vengono ormai lentamente sostituiti dagli "smart phones" (telefoni intelligenti), anche chiamati Personal Digital Assistant (PDA), proprio perchè in grado di svolgere funzioni eseguite in passato solo con un PC.

Questa crescita di mercato dei dispositivi mobili non è stata però supportata da una crescita comparabile nel software disponibile: la potenza di calcolo e di comunicazione è enorme comparata ad un vecchio PC degli anni 80, eppure non abbiamo che una frazione infinitesimale del software che avevamo sui nostri vecchi PC.

Perchè per i dispositivi mobili è disponibile così poco software?

Una delle ragioni risiede nel modello di sicurezza adottato dai dispositivi. Per chiarire questo punto consideriamo a titolo esemplificativo l'approccio usato dalla piattaforma per dispositivi mobili JAVA MIDP 2.0 (Mobile Information Device Profile). Il modello di sicurezza della piattaforma è basato su un "rapporto di fiducia": un'applicazione software viene eseguita sul dispositivo solo se la sua origine è in qualche modo certificata e ci si fida di tale certificazione. In estrema sintesi, l'applicazione è accettata solo se è provvista di una firma digitale conosciuta (relativa a chi ha sviluppato l'applicazione) e della quale ci si fida. Il livello di fiducia determina quindi i privilegi che l'applicazione avrà sul dispositivo, segregandola in un appropriato "dominio di fiducia".

Tale modello ha due problematiche principali. La prima è che una firma digitale può essere solamente accettata o rifiutata, ovvero l'interoperabilità in un dominio è totale o assente. E' possibile ad esempio proibire ad un'applicazione di connettersi ad Internet, ma non è possibile proibire che si connetta utilizzando uno specifico protocollo o solo ad un determinato dominio. Se un utente ha un servizio di pagamento disponibile sul proprio cellulare e scarica, ad esempio, un'applicazione per pagare il parcheggio, l'utente non potrà proibire a tale applicazione di pagare parcheggi per l'intera città. In sintesi, l'utente non può configurare le proprie preferenze di sicurezza e privacy per le applicazioni software che utilizza sul proprio dispositivo mobile.

La seconda problematica che caratterizza l'attuale modello di sicurezza utilizzato dai dispositivi mobili consiste nella mancanza di una "semantica" della firma digitale. Questo è un problema sia per gli sviluppatori delle applicazioni che per gli utenti finali.

Consideriamo il punto di vista degli utenti finali. Un utente è costretto ad accettare il software che scarica senza sapere in realtà quali garanzie in termini di sicurezza quell'applicazione potrà fornire. Ad esempio, un utente potrebbe fidarsi di SuperGame Inc ma potrebbe non voler scaricare ed eseguire giochi che non si fermano quando la batteria del suo dispositivo è ad un livello inferiore al 20%. Tale scelta al momento non è possibile.

Una precedente versione dell'articolo è apparsa su *Le Scienze Web News*, Mar. 08.

Prendiamo ora in esame il punto di vista dei produttori di software mobile. Uno sviluppatore non ha alcun modo di dichiarare quali azioni rilevanti per la sicurezza il suo software rispetterà, in modo tale che poi tali azioni siano controllabili automaticamente dal dispositivo dell'utente. Firmando il software dichiara solamente che il suo codice è sicuro, ma senza poter fornire in realtà alcuna garanzia concreta.

La principale conseguenza è che sviluppare un'applicazione per il mercato dei dispositivi mobili è un'operazione complessa e onerosa visto che gli sviluppatori devono convincere gli operatori che il loro software è sicuro e non farà niente di dannoso. La necessità di un "rapporto di fiducia" con l'operatore per poter sviluppare un servizio è quindi un grosso ostacolo per gli sviluppatori di software mobile. Infine, tale modello aumenta la responsabilità degli operatori per il software che sarà eseguito su dispositivi, in quanto unici responsabili dell'accettazione (in termini di fiducia) di tale codice.

2. L'approccio Security-By-Contract (SxC)

L'approccio Security-By-Contract (SxC), sviluppato nel contesto del progetto europeo S3MS (<http://www.s3ms.org>), rappresenta una risposta concreta al problema descritto della sicurezza delle applicazioni software per dispositivi mobili.

L'idea chiave può essere riassunta come segue: una firma digitale non dovrebbe certificare unicamente l'origine del codice (chi lo ha prodotto), ma fornire assieme al codice un "contratto" che descrive le caratteristiche salienti dell'applicazione in termini di sicurezza.

Come mostrato in Figura 1, il framework SxC è sostanzialmente formato da tre gruppi di attori principali: l'operatore della rete, l'utente e gli sviluppatori dei servizi.

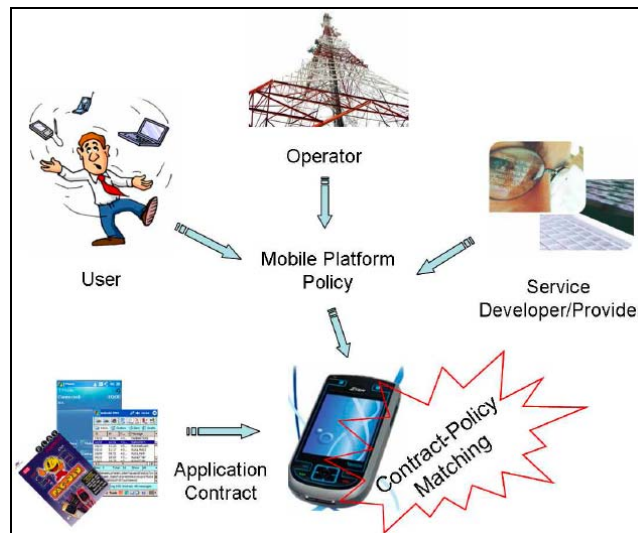


Figura 1: Attori principali dell'approccio SxC

Gli sviluppatori di software per dispositivi mobili devono fornire una descrizione del comportamento, in termini di sicurezza, che la loro applicazione seguirà una volta scaricata ed eseguita su una piattaforma mobile. Tale descrizione è chiamata **contratto**. Un contratto sarà quindi formato da una serie di regole come ad esempio:

- l'applicazione non invia più di un dato numero di messaggi SMS in una sessione
- l'applicazione si ferma se la batteria è inferiore a un certo livello
- l'applicazione utilizza solo determinati URL's

Firmando la propria applicazione software, uno sviluppatore può quindi certificare non solo l'origine del codice ma anche il comportamento, in termini di sicurezza, che l'applicazione seguirà una volta scaricata. In altre parole, certifica che l'applicazione rispetta le regole di sicurezza dichiarate nel contratto (che viene scaricato insieme all'applicazione).

Sia gli utenti che gli operatori sono invece interessati che tutte le applicazioni sviluppate per il proprio dispositivo mobile siano sicure, ovvero che non eseguano azioni dannose o maligne quando eseguite sul dispositivo. Con SxC tale controllo è possibile grazie alla definizione di una **policy (politica di sicurezza)**, ovvero una lista di azioni permesse sul dispositivo. Le policy presenti su un dispositivo possono essere definite dall'operatore della rete come dall'utente e possono anche essere specifiche di una singola applicazione, come

Una precedente versione dell'articolo è apparsa su *Le Scienze Web News*, Mar. 08.

nell'esempio seguente.

Esempio 1. Consideriamo di scaricare un'applicazione (gioco degli scacchi) sul nostro telefono cellulare. L'applicazione ha il seguente contratto:

- l'applicazione utilizza solo connessioni di rete HTTPS
- l'applicazione non spedisce messaggi SMS

Sul nostro dispositivo settiamo la seguente policy e l'assegnamo all'applicazione (i.e. l'applicazione dovrà rispettare la policy) selezionandola dall'elenco di policy disponibili (come mostrato in Figura 2):

- l'applicazione può usare solo connessioni di rete HTTP e HTTPS
- l'applicazione può spedire solamente 5 messaggi SMS



Figura 2: L'utente seleziona una policy per un'applicazione (gioco degli scacchi)

Una volta scelta la policy e assegnata all'applicazione, il dispositivo controllerà che il contratto dell'applicazione sia conforme con la policy: viene verificato che i requisiti di sicurezza dell'utente (definiti nella policy) siano rispettati dall'applicazione. In caso affermativo, il contratto rispetta la policy e l'applicazione può quindi essere eseguita in modo sicuro. Nell'esempio precedente è intuitivo verificare che il contratto rispetta la policy. Le regole nel contratto corrispondono alle azioni permesse dal dispositivo dichiarate nella policy.

Il paradigma SxC è presente in tutte le fasi dello sviluppo di un'applicazione mobile (Fig.3). Diverse tecniche possono essere utilizzate in queste fasi per garantire che un'applicazione rispetti effettivamente il proprio contratto o una polizza definita su un dispositivo mobile. Tali tecniche dipendono dalla fase nel life-cycle di sviluppo dell'applicazione. Per esempio, per verificare prima del download che un'applicazione rispetti le regole dichiarate nel contratto si possono utilizzare tecniche di verifica automatica basate su "analisi statica" del codice, mentre per monitorare l'esecuzione di un'applicazione si può utilizzare una tecnica chiamata "inline monitoring" (vedi Sezione 3).

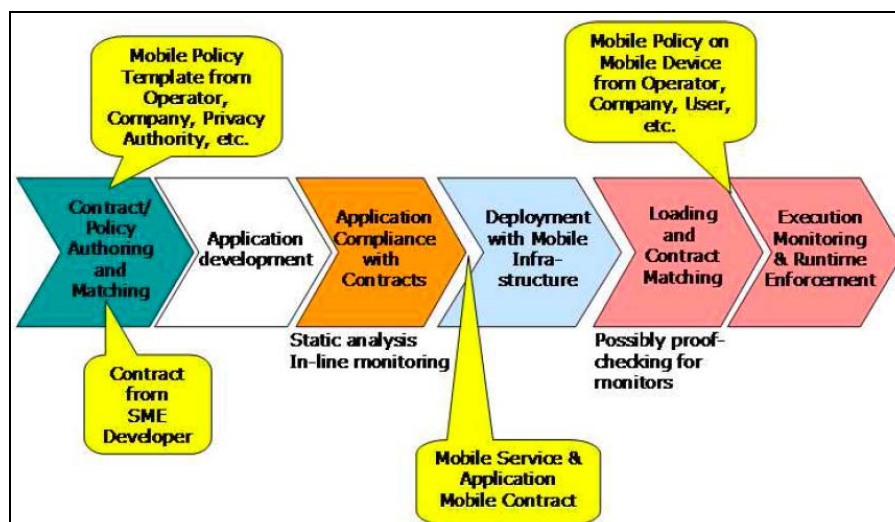


Figura 3: Life-cycle di un'applicazione o servizio SxC

In questo articolo non ci soffermeremo su ogni singola tecnica sviluppata, perchè richiederebbe molto spazio e dettagli tecnici. I lettori interessati sono invitati a consultare il sito Web del progetto S3MS (<http://www.s3ms.org>) dove è possibile scaricare tutti i documenti tecnici e le pubblicazioni scientifiche relative al paradigma SxC.

E' importante enfatizzare che l'approccio SxC non richiede la completa sostituzione degli attuali meccanismi di sicurezza, ma al contrario si è posto l'obiettivo di migliorarli.

3. Monitorare un'Applicazione con la Tecnica di Inline Monitoring

Cosa succede se scarichiamo un'applicazione che non ha un contratto? Possiamo eseguirla in modo sicuro? Oppure l'approccio SxC richiede che tutte le applicazioni abbiano necessariamente un contratto? E cosa succede se il contratto di un'applicazione non rispetta la policy del dispositivo mobile?

L'approccio SxC gestisce tutte queste situazioni utilizzando una tecnica chiamata **inline monitoring**. Per mezzo di questa tecnica è possibile far rispettare una policy da un'applicazione inserendo opportuni controlli nel codice dell'applicazione. In questo modo un utente può scaricare qualsiasi applicazione software (anche sprovvista di un contratto) ed eseguirla in modo sicuro nel proprio dispositivo. La tecnica di inline monitoring sviluppata nel contesto di SxC viene eseguita direttamente sul dispositivo e quindi non dipende da nessun servizio sviluppato da terze parti. Il risultato è che ogni azione rilevante per l'utente in termini di sicurezza (ovvero ogni azione dichiarata nella policy) viene automaticamente controllata durante l'esecuzione dell'applicazione.

Esempio 2. Consideriamo due utenti che giocano con l'applicazione introdotta nell'esempio 1 (gioco degli scacchi) utilizzando i loro telefon (Figura 4). Assumiamo che tale applicazione non abbia un contratto oppure che il contratto non rispetti la policy dei due telefoni. Nel PDA di sinistra non c'è alcun sistema di inline monitoring, quindi l'applicazione è eseguita senza alcun controllo di sicurezza. Nel PDA di destra invece è installato un sistema di inline monitoring che tra le varie azioni controllate verifica che l'applicazione non spedisca più di un certo numero di messaggi (per non far spendere troppo all'utente). Quando si verifica la violazione (cfr Figura) il sistema di monitoring se ne accorge e l'azione incriminata non verrà eseguita ed il problema segnalato all'utente. L'utente potrà scegliere se visualizzare i dettagli della violazione o chiudere direttamente l'applicazione.

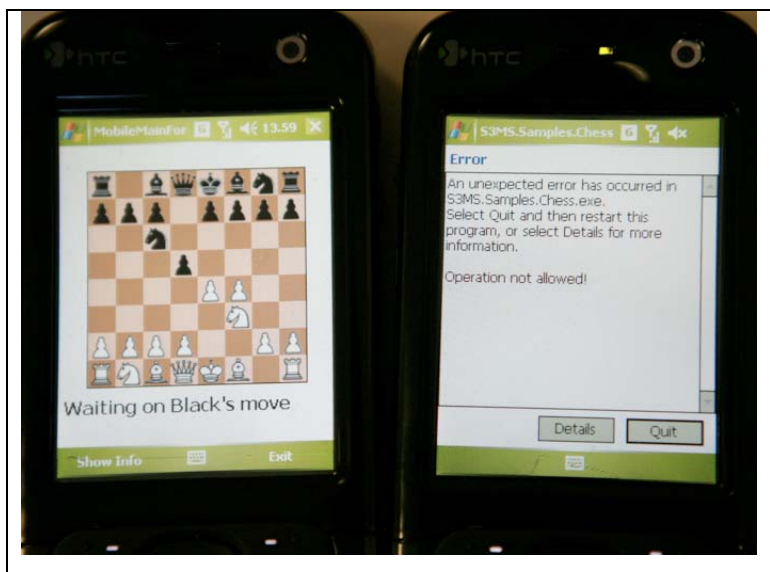


Figura 4: Due cellulari che giocano a scacchi. Il cellulare di destra cattura la violazione di una policy.

Lo scenario seguente illustra un ulteriore esempio di possibile utilizzo del servizio, con alcuni dettagli su ciò che avviene realmente e in maniera del tutto trasparente per l'utente.

Scenario 1. *Un utente definisce la propria policy o, se non si sente abbastanza competente, seleziona una policy predefinita dall'operatore. La policy selezionata è formalizzata nel dispositivo in un opportuno linguaggio logico. Questo permette al dispositivo di compilare la policy in uno specifico formato che verrà poi utilizzato nella fase di inline monitoring. Tale operazione avviene completamente in automatico, ovvero l'utente non è coinvolto (non è necessario che l'utente conosca un linguaggio tecnico). A questo punto la policy è pronta per essere utilizzata. Ipotizziamo ora che l'utente scarichi un'applicazione, ad esempio un gioco di scacchi, che non ha un contratto e quindi non può avvenire il controllo automatico tra la policy del dispositivo e il contratto dell'applicazione. Ma all'utente piace molto questa applicazione e vuole installarla comunque. Per eseguirla in modo sicuro, l'utente seleziona da un elenco una policy da applicare all'applicazione, come mostrato precedentemente in Figura 2. Una volta selezionata la policy, il dispositivo modifica l'applicazione inserendo tutti i controlli dichiarati nella policy (fase di inline monitoring). A questo punto l'applicazione è pronta per essere eseguita in modo sicuro: ogni volta che l'applicazione violerà una delle regole della policy dell'utente, il sistema non permetterà l'esecuzione dell'azione e segnalerà la violazione all'utente. Tale segnalazione può avvenire in modo più o meno drastico a seconda di come l'applicazione gestisce queste eccezioni di sicurezza. Per esempio, se l'applicazione non gestisce tali eccezioni, allora l'applicazione terminerà bruscamente. Se invece l'applicazione prevede la gestione delle eccezioni di sicurezza, l'applicazione potrebbe ad esempio terminare solo dopo aver segnalato all'utente cosa è successo.*

La figura successiva ci descrive in estrema sintesi i possibili flussi delle operazioni al momento dell'installazione e successiva esecuzione di un'applicazione.

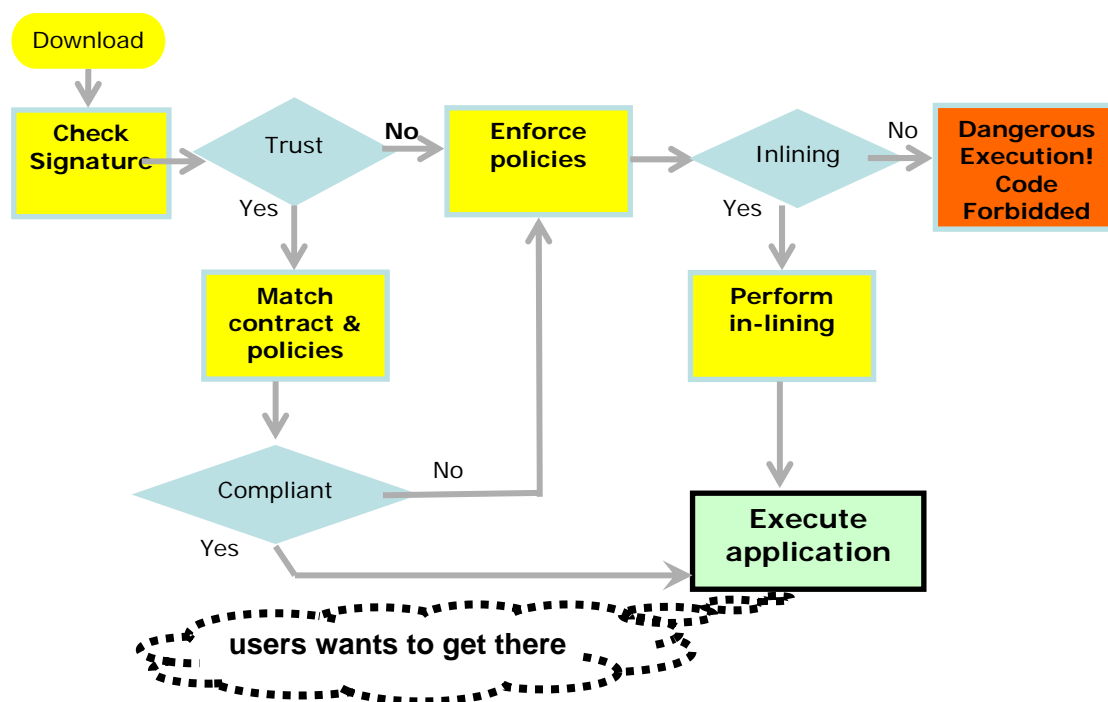


Figura 5: Cosa succede con SxC quando si scarica un'applicazione.

4. Conclusioni

In questo articolo è stato introdotto il paradigma Security-By-Contract (SxC), un approccio per migliorare la sicurezza nei dispositivi mobili sviluppato nel contesto del progetto europeo S3MS (<http://www.s3ms.org>). L'intuizione di base è che un'applicazione software viene scaricata insieme ad un "contratto" che descrive tutte le interazioni rilevanti, in termini di sicurezza, che l'applicazione avrà col dispositivo mobile sul quale verrà eseguita. Il contratto dovrà essere quindi accettato dal dispositivo in funzione di una policy di sicurezza definita dall'utente o dall'operatore della rete. Tale paradigma non sostituirà gli attuali sistemi di sicurezza ma li migliorerà, fornendo un meccanismo di sicurezza flessibile, semplice e scalabile per i dispositivi mobili del futuro.

Riferimenti

Documenti tecnici e pubblicazioni scientifiche possono essere liberamente scaricate dal sito Web del progetto S3MS: <http://www.s3ms.org>.