Divenire-software delle reti e Cyber Security

Author : Andrea Boggio

Date : 4 Novembre 2019

Il software si è mangiato il mondo

Nel 2011 Mark Andreessen, imprenditore informatico statunitense e co-fondatore, tra le varie cose, di Netscape, scrisse un famoso articolo per il *Wall Street Journal* dal titolo "*Why Software Is Eating The World*". All'alba degli anni 10 del terzo millennio, Andreessen evidenziava che in quello specifico momento storico era in atto un cambiamento epocale dal punto di vista tecnologico ed economico, con le *software company* in procinto di conquistare enormi quote di mercato.

La previsione era che, nell'arco di un decennio, moltissimi settori dell'industria sarebbero stati scompaginati dal software per due ragioni fondamentali:

- tutta la tecnologia software necessaria a trasformare radicalmente i processi di produzione e distribuzione di beni e servizi era finalmente funzionante e disponibile su scala globale;
- il software si era già insinuato nella catena del valore di industrie che erano sempre e solo state percepite come facenti parte del mondo fisico.

Per esemplificare la portata del cambiamento in atto, Andreessen faceva una serie di contrapposizioni tra *software company* e aziende tradizionali che sarebbero diventate, nel corso degli anni successivi, veri e propri *topos*: Amazon VS le maggiori catene di librerie, Netflix VS Blockbuster, iTunes, Spotify e Pandora VS le tradizionali etichette musicali, Pixar VS case di produzione cinematografica, Snapfish e Flickr VS Polaroid, Skype VS le compagnie telefoniche tradizionali. Non solo: anche i settori finanziario, sanitario ed energetico, nonché colossi quali FedEx e Wal-Mart, avevano già "riversato" interamente all'interno del software alcuni processi strategici produttivi, distributivi e logistici.

A distanza di quasi 10 anni dall'articolo di Andreessen la dinamica descritta si è ulteriormente accentuata: oggi le maggiori *software company* hanno differenziato le proprie attività aggredendo nuovi business. Amazon sfida Wal-Mart nel commercio al dettaglio, Google investe

nei settori dei servizi alle imprese, del trasporto, della salute e dell'energia, Facebook punta ai servizi finanziari lanciando addirittura Libra, la propria criptovaluta.

Il processo attraverso cui il software "*si mangia il mondo*", che preferisco generalizzare con l'espressione *divenire-software,* rientra nella più ampia dinamica di *Digital Transformation* che investe ormai da diversi anni il pianeta determinando notevoli cambiamenti a livello sociale, culturale, organizzativo e politico.

Dinamiche di sviluppo delle reti

Nel contesto generale appena descritto, l'universo delle tecnologie dell'informazione e della comunicazione (ICT) rappresenta simultaneamente il motore dell'innovazione e il campo di sperimentazione e applicazione naturale della trasformazione digitale.

Le infrastrutture di comunicazione che abilitano i servizi digitali sono la spina dorsale – il *backbone* – che permette alla società globale di attuare lo spostamento nella sfera immateriale di relazioni, beni e servizi finora esclusivamente radicati nel dominio del mondo fisico. Ne consegue che le dimensioni classiche della sicurezza dell'informazione – la disponibilità, la riservatezza e l'integrità – diventano elementi di necessaria garanzia nella generale prospettiva di resilienza, affidabilità e prosperità della nostra società.

Gli operatori di comunicazione, con particolare riferimento a coloro che offrono servizi di comunicazione elettronica, rivestono un ruolo fondamentale nell'erogazione di servizi essenziali alla comunità, come sancito anche dal quadro normativo a livello nazionale e internazionale: proprio alla luce del loro ruolo e della loro intrinseca sensibilità, per tali operatori sono previste specifiche misure di cyber security e obblighi formali verso le istituzioni.

Come noto, la grande narrazione che oggi viene promulgata a vari livelli sotto il nome generico di **5G** comporterà nuovi modelli di business, nuovi scenari di utilizzo e impatti la cui portata complessiva è ancora difficile da prevedere con precisione, come ho avuto modo di illustrare in un precedente articolo. La visione 5G non può essere soddisfatta attraverso gli approcci di progettazione di rete convenzionali: entra in gioco una quantità di innovazione mai vista prima e gli operatori di comunicazione dovranno utilizzare nuove tecnologie per consegnare servizi 5G sicuri e senza soluzione di continuità attraverso confini di rete multipli. Per ottenere l'interoperabilità, le reti 5G dovranno interconnettersi con soluzioni di rete *legacy* e diverse tecnologie di accesso.

Per raggiungere gli obiettivi tecnici le reti 5G devono essere altamente flessibili, scalabili e programmabili. Il trend tecnologico che abiliterà a breve questa rivoluzione è profondamente segnato dal *divenire-software* delle reti di comunicazione e si può riassumere in alcuni elementi fondamentali:

- virtualizzazione delle architetture di rete attraverso *Software Defined Network* (SDN). La SDN permette di programmare dinamicamente la rete disaccoppiando il processo di *forwarding* dei pacchetti (*Data Plane*) da quello di *routing* (*Control Plane*) tramite elementi di controllo centralizzati, intelligenti e residenti nel Cloud;

- virtualizzazione delle funzioni di rete, meglio conosciuta come *Network Function Virtualization* (NFV). NFV permette configurazioni di rete multiple tra cui, per esempio, l'allocazione dinamica di indirizzi IP, il *firewalling* e la scalabilità della rete in tempo reale;
- diffusione delle architetture distribuite quali, ad esempio, il *Multi-access Edge Computing* (MEC). MEC porta i dati vicino all'utente finale, spostando le funzioni di calcolo e *storage* ai bordi (*edge*) della rete. Questo scenario riduce notevolmente i tempi di latenza e il volume dei dati gestiti dagli elementi centrali della rete (*core* e *backbone*), apportando benefici in termini di aumento delle prestazioni;
- *Distributed Core Network*, ovvero l'abilità di spostare le funzioni di rete *core* laddove è maggiore la richiesta di utilizzo. La distribuzione delle funzioni (quali, ad esempio, l'autenticazione) introduce vantaggi prestazionali abbassando i tempi di latenza e permettendo le comunicazioni a latenza ultra-bassa;
- *Network Slicing*, ovvero la capacità di eseguire reti virtuali multiple su una singola infrastruttura di rete condivisa. Lo *Slicing* della rete abbraccia diversi livelli della pila ISO/OSI: interfacce radio, VLAN *Layer 2* e *routing/forwarding* virtuale *Layer 3*;
- introduzione dei modelli architetturali basati su *container* e micro-servizi;
- adozione massiccia di interfacce *Open*.

Sicurezza del software: croce e delizia

È evidente che il *divenire-software* è una dinamica inarrestabile e pervasiva che nella propria traiettoria disegna un percorso ormai irreversibile in cui il mondo fisico e quello virtuale, ibridandosi, tendono a confondersi. I contorni si fanno sempre più sfumati e liquidi, il volume dei dati assume dimensioni abnormi (*Big Data*) e il software stesso necessita di evoluzioni continue (Intelligenza Artificiale).

Tutte le *mirabilia* rese possibili dalla *Digital Transformation* si fondano sul software: peccato che, ad oggi, **non c'è modo di scrivere software intrinsecamente sicuro**.

Come giustamente sostiene Bruce Schneier, il software è spesso scritto male (*poorly written*) perché, a parte poche eccezioni, il mercato non premia il software di buona qualità: "*economico*" e "*veloce sul mercato*" sono attributi più importanti della qualità e, del resto, per la maggior parte di noi il software scritto male è in ogni caso di qualità sufficiente e accettabile.

Questa filosofia ha permeato l'industria a tutti i livelli: le aziende non premiano la qualità del software allo stesso modo in cui premiano la capacità di consegnare un manufatto nel rispetto dei costi e dei tempi previsti. Le università si focalizzano più su un codice funzionante che su un codice affidabile. La maggior parte dei consumatori non intende pagare il sovrapprezzo determinato dal miglioramento della qualità del software.

Il software moderno è costellato di *miriadi* di *bug*. Alcuni sono inerenti alla complessità del codice stesso, ma altri sono errori di programmazione. Molti *bug* non sono corretti durante il processo di sviluppo e rimangono nel prodotto finito anche dopo che questo viene completato e immesso sul mercato. Alcuni *bug* sono anche vulnerabilità di sicurezza e alcune di queste possono essere sfruttate da attaccanti e malintenzionati.

Chiaramente non tutti i processi di sviluppo sono uguali: Microsoft ha impiegato risorse enormi per migliorare nel tempo il proprio processo di sviluppo al fine di minimizzare il numero di vulnerabilità di sicurezza presenti nei propri prodotti, che non sono perfetti – in quanto la perfezione va oltre le capacità tecnologiche attualmente disponibili – ma sono superiori alla media. Apple è famosa per la qualità del proprio software, lo stesso vale per Google. Alcuni piccoli pezzi di codice sono di qualità estremamente elevata, ad esempio il software dell'avionica dei velivoli.

Internet of Things (IoT) significa anche la proliferazione di oggetti e dispositivi più o meno intelligenti dotati di specifico software (*miliardi* di oggetti): più linee di codice, più *bug*, più vulnerabilità di sicurezza. Il costo di tali dispositivi tende a essere contenuto al massimo e ciò significa utilizzare programmatori poco esperti, processi di sviluppo software spesso superficiali e fare ampio riutilizzo di codice, con il concreto rischio di effetti domino incontrollabili (una singola vulnerabilità presente nello stesso frammento di codice utilizzato in prodotti diversi ne determina la replica immediata su scala maggiore).

Il software da cui dipendiamo – quello che gira sui nostri computer e telefoni, nelle nostre automobili e nei dispositivi medicali, su Internet, nei sistemi di controllo delle nostre infrastrutture critiche, sulle reti di nuova generazione (5G) – è insicuro in molteplici modi. Il problema non è solo trovare tutte le vulnerabilità e correggerle: ce ne sono troppe e la maggior parte di esse non viene nemmeno mai scoperta. È un problema di correzione, nel tempo, delle vulnerabilità note attraverso opportuni processi di *patching*.

Un veloce *excursus* sulla storia delle vulnerabilità software ci porta dalla nascita della *responsible disclosure* fino agli attuali programmi di *bug bounty*, con il naturale approdo al *patching* come possibile soluzione – *ex post* - dei problemi di insicurezza del codice. Peccato che, anche in questo caso, sempre secondo Schneier, il *patching* come paradigma di sicurezza stia fallendo:

- molte volte il rilascio delle *patch* di sicurezza non è tempestivo. Nel caso, ad esempio, di *smartphone* con sistema operativo Android questo significa che l'esposizione delle vulnerabilità non corrette sui dispositivi può rimanere tale diversi mesi;
- nel caso di sistemi particolarmente datati, le *patch* non sono più rilasciate e i sistemi rimangono vulnerabili nel tempo;
- a volte le *patch* che correggono vulnerabilità software compromettono la stabilità dei sistemi in cui sono installate, con conseguenti *rollback* (nei casi migliori) o *brick* degli stessi (nei casi peggiori);
- nei dispositivi a basso costo e sistemi *embedded* (tanti router Internet domestici, dispositivi IoT, webcam, etc) le *patch* devono essere scaricate e installate manualmente dagli utenti in assenza di un processo remoto e automatico. Tale mancanza di automatismo genera, normalmente, assenza di consapevolezza e delle conseguenti azioni di rimedio da parte dell'utente medio.

Gli attuali sistemi di *patching* saranno sempre meno adeguati in un mondo caratterizzato dal *divenire-software*, in cui i computer e il software vengono incorporati in qualsiasi cosa. Secondo Schneier una strada possibile è l'integrazione dei due principali paradigmi adottati

dall’industria dello sviluppo software: *Waterfall* e *Agile*. Nel tradizionale modello *Waterfall* l’iter naturale dello sviluppo del codice prevede la definizione dei requisiti, delle specifiche, la progettazione, l’implementazione, la verifica e la manutenzione. Il modello *Agile*, più recente, funziona diversamente: come prima cosa si costruisce un prototipo per soddisfare le esigenze basilari del cliente, si osservano gli errori e i fallimenti, si ripara velocemente, si aggiornano i requisiti e le specifiche e si ripete tutto il processo più volte nel tempo. Non riusciamo a ingegnerizzare la sicurezza al 100% sin dall’inizio, quindi non abbiamo altra scelta che rilasciare e applicare le *patch* velocemente. A causa dell’intrinseca complessità di Internet e dei dispositivi a essa connessi, abbiamo bisogno sia della stabilità a lungo termine del modello *Waterfall* sia della capacità reattiva del modello *Agile*.

Conclusioni

Le reti di quinta generazione hanno ottime caratteristiche di resilienza, disponibilità e affidabilità. La flessibilità di questo modello è rappresentata dalla possibilità, per i servizi, di condividere dinamicamente molteplici funzioni di rete, che a loro volta condividono dinamicamente diversi micro-servizi di base. Le componenti a maggiore **criticità** sono relative al *core* della rete, all’interfacciamento tra le reti e ai sistemi di gestione e supporto: ancora una volta si tratta di **elementi software**.

Il cambio di paradigma introdotto dal *divenire-software* delle reti di nuova generazione è senz’altro positivo in termini di:

- gestione (maggiore visibilità determinata da un modello di controllo centralizzato);
- possibilità di applicare la *Security by Design*: le reti di nuova generazione includono funzioni e *capability* di sicurezza native che possono essere progettate e implementate sin da subito e non, come storicamente accaduto, *ex post* (prima i *router*, poi i *firewall*);
- protezione avanzata: essendo i tempi di *provisioning* e *remediation* estremamente ridotti, la risposta agli attacchi può essere parzialmente automatizzata e variare dinamicamente anche durante un evento di sicurezza. Inoltre, gli aggiornamenti e le *patch* di sicurezza possono essere facilmente programmati ed eseguiti;
- *Defense in Depth*: i dati possono essere efficacemente protetti tramite l’integrazione delle diverse funzioni di sicurezza disponibili ad ogni livello;
- aumento complessivo delle capacità di risposta agli incidenti;
- maggiore resilienza agli attacchi DDoS;
- *Zero Trust*: possibilità di eliminare il *trust* debole segmentando l’accesso alla rete e autorizzando, ispezionando e cifrando tutto il traffico in modalità *end-to-end*.

Non esistono risposte semplici alla domanda di sicurezza generata da uno scenario di tale complessità, ma è opportuno, almeno, cercare di imparare le lezioni del passato facendone tesoro:

- il processo di *Secure Software Developement Lifce Cycle* deve essere opportunamente integrato sia nel paradigma *Waterfall* sia in quello *Agile*, focalizzando gli sforzi sia sulle fasi *ex-ante* (definizione dei requisiti di sicurezza, progettazione delle interfacce, *Static Application Security Testing*, etc) sia su quelle *ex-post (Vulnerability Assessment,*

Penetration Test, Dynamic Application Security Testing, *Patching*, etc);

- per mitigare il rischio di compromissione della *Supply Chain*, è necessario implementare processi di selezione dei propri fornitori con particolare attenzione ai requisiti di sicurezza di questi ultimi. È opportuno richiedere l'utilizzo di standard e *best practice*, unitamente al possesso delle opportune certificazioni di qualità e sicurezza con le relative evidenze. I fornitori sono coloro che, materialmente, scrivono il codice su cui si basano le reti di nuova generazione degli operatori di comunicazione;
- per mitigare i rischi derivanti dalle vulnerabilità di determinati protocolli di comunicazione e segnalazione, è necessario attuare misure tecniche di sicurezza che prevedano il monitoraggio continuo e la protezione attiva dei punti di interconnessione verso le reti di terze parti. Le reti di nuova generazione dispongono di *capability* adeguate a gestire tempestivamente situazioni di emergenza di questo tipo.

Il *divenire-software* delle reti si innesta all'interno di un'ampia trasformazione digitale della società operante a vari livelli ed eredita le caratteristiche tipiche del software. Al netto delle *capability* di modularità, scalabilità e programmabilità delle funzioni, un punto di attenzione è rappresentato dalla storica difficoltà di garantire un adeguato livello di sicurezza del software.

Nota

Ho utilizzato per la prima volta la locuzione *Divenire-software* durante un evento organizzato da AFCEA dal titolo *"Hybrid Landscape: strategie e sistemi per la difesa e sicurezza delle reti"*.
Slide:
https://www.afcearoma.it/images/icagenda/files/Presentazioni_Eventi/Anno2019/19_giugno_2019/Convegno_AFCEA_Andrea_Boggio.pdf
Video: https://youtu.be/pNZSsDlPyqQ

Articolo a cura di **Andrea Boggio**