smeup

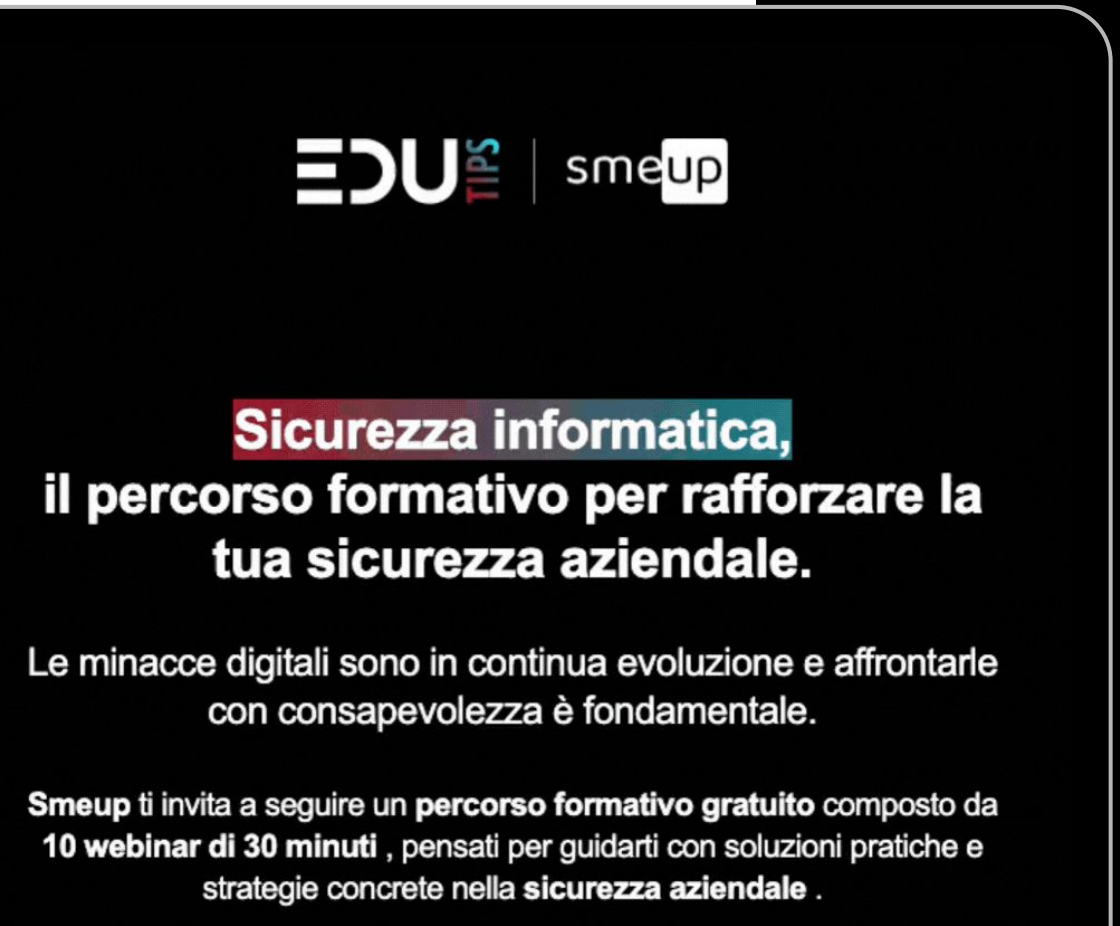
WEBINAR

Sicurezza delle applicazioni

Proteggi le tue applicazioni dalle vulnerabilità più comuni.

8 LUGLIO
2025

EDU TIPS
CYBERSECURITY



EDU TIPS Cybersecurity

10 webinar TIPS - 30 minuti al mese
su tematiche specifiche di Cybersecurity

DATA	DESCRIZIONE
28/1	Mese 1: **Sicurezza dell'infrastruttura**
13/2	Mese 2: **Cyber Security Awareness**
13/3	Mese 3: **Data Protection**
10/4	**Mese 4: Identità e accesso**
13/5	Mese 5: **Cloud security**
12/6	**Mese 6: Incident response**
8/7	**Mese 7: Sicurezza delle applicazioni**
11/9	Mese 8: **Analisi delle minacce e vulnerability assessment**
10/10	Mese 9: **Sicurezza delle reti industriali (OT)**
11/11	Mese 10: **Sicurezza in ambito A.I.**

👉 SCOPRI DI PIÙ

EDU TIPS

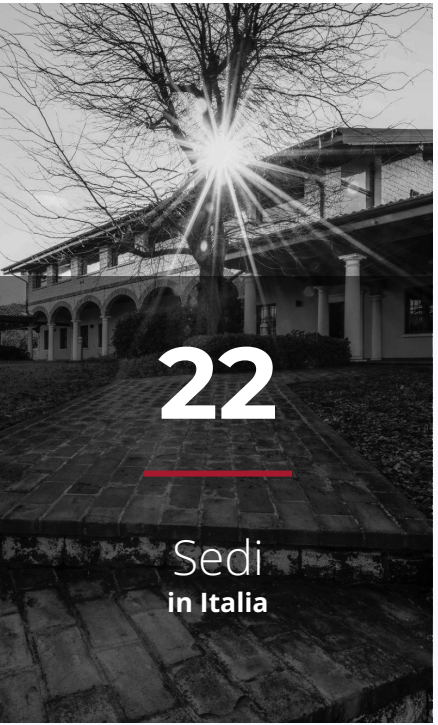
WEBINAR
SPECIALE CYBER SECURITY

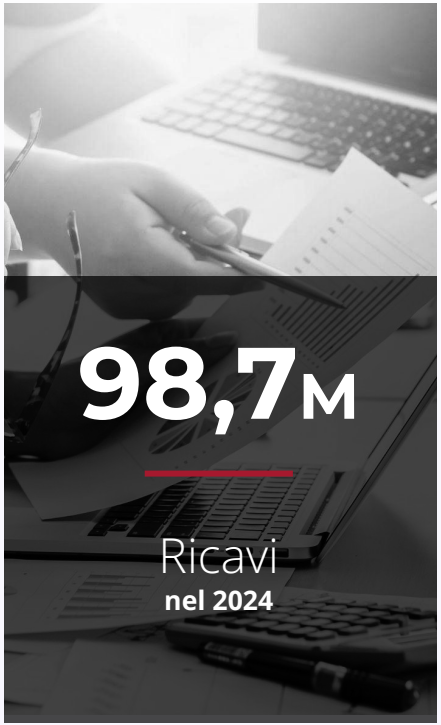
OLIVIERO
MAESTRELLI

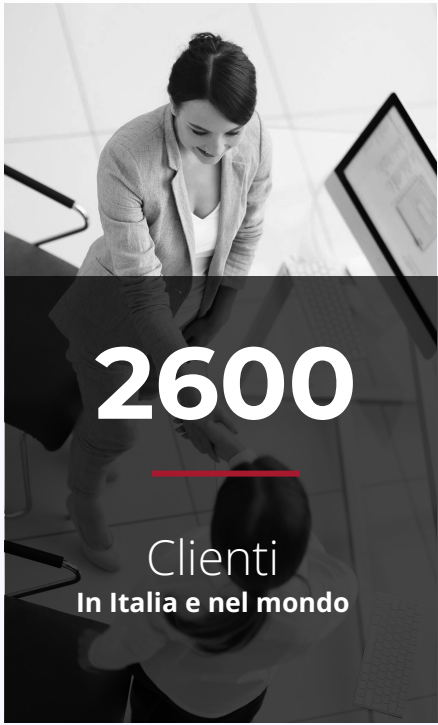
R&D @smeupLAB

COMPANY **OVERVIEW**

smeup in Numeri.







BUSINESS SECTOR

BUSINESS SOFTWARE APPLICATION

Soluzioni **Software** per **PMI** e **grandi industrie.**

Ogni azienda è unica. smeup lo sa!
Usare la digitalizzazione per sviluppare il business e generare valore, ***insieme***.

- **e** GESTIONALI ERP
- **b** BUSINESS ANALYTICS
- **d** DOCUMENTALE
- **w** WEB & MOBILE APPLICATION
- **f** IOT E INTEGRAZIONE INDUSTRIALE
- **l** LOGISTICA E TRASPORTI

BUSINESS SECTOR

INFRASTRUCTURE, CLOUD & SECURITY

Soluzioni per **Architetture IT** e **servizi gestiti.**

Innovazione e **sicurezza** per rispondere ai bisogni delle aziende.

- i INFRASTRUTTURA
- c CLOUD
- cs CYBER SECURITY
- ps IBM POWER SYSTEMS

SICUREZZA DELLE **APPLICAZIONI**

L'importanza della sicurezza delle applicazioni.

Come garantire che il software che sviluppiamo e utilizziamo sia robusto e protetto dalle crescenti minacce informatiche?

La sicurezza non è solo un aspetto tecnico, ma una responsabilità che attraversa ogni fase del ciclo di vita del software, dal concepimento alla dismissione. Ignorare questa priorità può portare a rischi significativi, sia in termini economici che reputazionali.

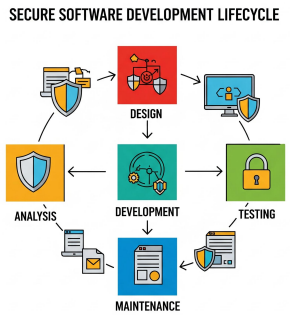
- La sicurezza informatica è una responsabilità condivisa che trascende la mera implementazione tecnica, coinvolgendo processi, persone e cultura aziendale.
- Ogni fase del software può introdurre nuovi rischi.
- Il software obsoleto è una vulnerabilità comune.
- I cyber attacchi sfruttano le debolezze note.

Il ciclo di vita dello sviluppo sicuro del software.

Tradizionalmente, la sicurezza veniva spesso considerata un'aggiunta 'a posteriori' nello sviluppo software. L'esperienza ci ha però dimostrato che un tale approccio è inefficace e costoso.

Per costruire applicazioni veramente resilienti, è fondamentale incorporare la sicurezza in ogni singola fase del ciclo di vita dello sviluppo. Questo approccio, noto come **Secure Software Development Lifecycle (SSDLC)**, ci permette di identificare e mitigare i rischi in modo proattivo, riducendo le vulnerabilità e garantendo un prodotto finale più affidabile.

- Analisi dei requisiti e identificazione delle minacce.
- Progettazione sicura e valutazione dei rischi.
- Sviluppo, test e distribuzione attenti alla sicurezza.
- Manutenzione e dismissione sicura del software.



Vulnerabilità comuni nelle applicazioni.

Tradizionalmente, la sicurezza veniva spesso considerata un'aggiunta 'a posteriori' nello sviluppo software. L'esperienza ci ha però dimostrato che un tale approccio è inefficace e costoso.

Per costruire applicazioni veramente resilienti, è fondamentale incorporare la sicurezza in ogni singola fase del ciclo di vita dello sviluppo. Questo approccio, noto come Secure Software Development Lifecycle (SSDLC), ci permette di identificare e mitigare i rischi in modo proattivo, riducendo le vulnerabilità e garantendo un prodotto finale più affidabile.

- Errori di programmazione e codice non sicuro espongono a rischi.
- Utilizzo di librerie di terze parti non verificate e non aggiornate causano potenziali falle.
- Credenziali e segreti mal gestiti aumentano le minacce.
- Ambienti di test trascurati creano nuove vulnerabilità.
- La mancanza di aggiornamenti post-rilascio è critica.
- **OWASP (Open Worldwide Application Security Project) Top 10.**

SICUREZZA DELLE **APPLICAZIONI**

Software obsoleto: un rischio per la sicurezza.

Nel mondo dinamico della tecnologia, il software è in continua evoluzione. Quello che è all'avanguardia oggi, può diventare obsoleto domani. Ma l'obsolescenza non è solo una questione di prestazioni o funzionalità; rappresenta una delle maggiori e più insidiose minacce alla sicurezza.

- Sfruttare le vulnerabilità note del software obsoleto è comune.
- I software meno recenti non ricevono patch.
- I sistemi obsoleti sono punti di accesso privilegiati per gli attaccanti.
- Aggiornamenti costanti riducono i rischi.

SICUREZZA DELLE **APPLICAZIONI**

L'impatto economico del software obsoleto: il caso WannaCry.

Le vulnerabilità del software obsoleto non sono solo rischi teorici; hanno un costo molto reale e spesso devastante. Oltre al potenziale danno alla reputazione e alla perdita di fiducia dei clienti, le violazioni della sicurezza possono comportare spese ingenti per la remediation, multe normative e interruzioni operative.

L'attacco WannaCry ha dimostrato come la mancata applicazione di semplici aggiornamenti possa portare a perdite finanziarie miliardarie a livello globale.

- Wannacry ha sfruttato vulnerabilità note di EternalBlue.
- Microsoft aveva già rilasciato la patch necessaria.
- Sistemi non aggiornati hanno causato gravi danni operativi, reputazionali e finanziari.
- L'attacco ha provocato miliardi di danni globali.
- **200.000 sistemi** in più di **150 paesi**.

Manutenzione continua: la chiave per la sicurezza a lungo termine.

La sicurezza delle applicazioni non è un traguardo da raggiungere una volta per tutte, ma un processo continuo e dinamico.

La manutenzione, in questo contesto, non si limita alla correzione di bug, ma diventa un'attività strategica e incessante di monitoraggio, aggiornamento e adattamento per anticipare e neutralizzare le minacce emergenti. È un impegno costante che garantisce la longevità e l'integrità del software.

- Le patch e gli aggiornamenti sono continui.
- Monitorare costantemente le vulnerabilità esistenti con l'uso di strumenti di scansione e monitoraggio.
- Aggiornare regolarmente tutte le dipendenze.
- Il software richiede manutenzione costante.

Gestire il rischio di deprecazione delle funzioni e librerie.

Oltre alle vulnerabilità note del software obsoleto, esiste un rischio più subdolo ma altrettanto pericoloso: quello derivante da funzioni e librerie deprecate.

Queste componenti, pur essendo ancora presenti nel codice, non sono più supportate attivamente dagli sviluppatori e possono diventare silenziose fonti di problemi di sicurezza o di malfunzionamento. Ignorare la deprecazione può portare a incompatibilità inattese con nuovi aggiornamenti, errori imprevedibili e, soprattutto, a lacune nella sicurezza che gli attaccanti potrebbero sfruttare. È quindi essenziale avere una chiara strategia per gestire e modernizzare queste parti del codice.

- Funzioni e librerie deprecate introducono rischi silenziosi.
- Il software può smettere di funzionare dopo aggiornamenti.
- Inventario software e dipendenze è fondamentale.
- Pianificare alternative e refactoring per tempo.
- **Software Bill Of Material**.

SICUREZZA DELLE **APPLICAZIONI**

L'importanza cruciale dell'aggiornamento delle dipendenze.

Raramente sviluppiamo software completamente da zero; ci affidiamo a librerie e componenti di terze parti, spesso open source. Questa pratica, pur accelerando lo sviluppo, introduce un punto di potenziale vulnerabilità se tali dipendenze non vengono gestite e aggiornate con rigore. La sicurezza del nostro software è intrinsecamente legata alla sicurezza di ogni singolo pezzo di codice che utilizziamo.

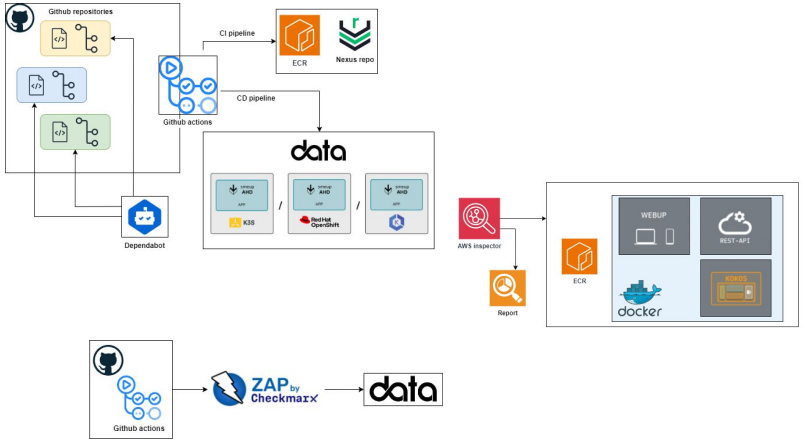
- L'open source offre trasparenza e flessibilità, ma richiede la stessa attenzione alla sicurezza e agli aggiornamenti delle soluzioni proprietarie.
- Verificare e aggiornare le dipendenze esterne.
- **Log4Shell** esempio di vulnerabilità diffusa.
- Manutenzione costante riduce i rischi.

SICUREZZA DELLE **APPLICAZIONI**

Strategie e buone pratiche per una sicurezza robusta.

Identificare i problemi è solo il primo passo. Per costruire un ecosistema software veramente sicuro, è essenziale adottare proattivamente una serie di buone pratiche e integrarle nel nostro quotidiano.

- Progettazione sicura fin dalle prime fasi.
- Integrazione della sicurezza nei processi di sviluppo.
- Gestire aggiornamenti e fine vita del software.
- Formazione continua del personale sullo sviluppo sicuro.
- Automazione dei test di sicurezza
- **KPI (Mean Time To Detection, Mean Time To Remediate)**



Sicurezza negli ambienti legacy.

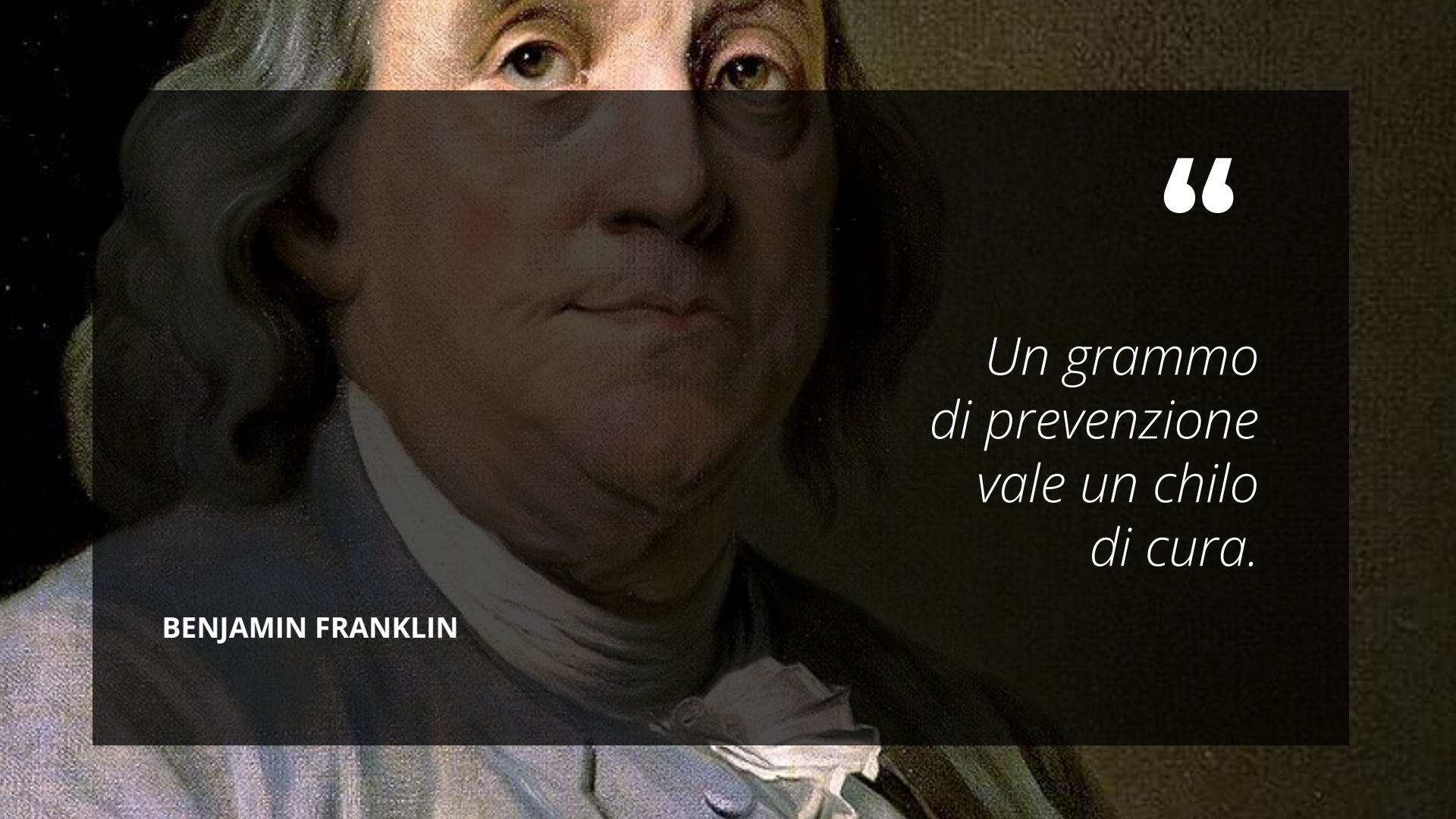
Molte organizzazioni si trovano a gestire una parte significativa della loro infrastruttura basata su software legacy, sistemi più datati che spesso non erano stati progettati con le attuali esigenze di sicurezza in mente. Questi ambienti presentano sfide uniche e possono diventare anelli deboli nella catena di sicurezza complessiva.

- Mappare software e dipendenze attuali.
- Isolare componenti legacy per ridurre rischi segmentando la rete o utilizzando container/VM dedicate.
- Aggiornare gradualmente con approccio incrementale.

La sicurezza come investimento strategico e non un costo.

La sicurezza del software è parte integrante della salute digitale di un'organizzazione. Avere software aggiornato e politiche di aggiornamento chiare, ambienti di test, osservabilità dell'installato, e dell'utilizzato non sono un lusso, ma una necessità.

- Software aggiornato significa maggiore sicurezza e maggiore resilienza operativa.
- Coinvolgere il management è cruciale per ottenere risorse, supporto e allineamento strategico.
- Agire subito riduce i rischi e i costi.
- L'aggiornamento è un investimento essenziale in termini di continuità aziendale, reputazione e conformità.



Un grammo di prevenzione vale un chilo di cura.

BENJAMIN FRANKLIN

Domande chiave per la riflessione e la discussione.

- Secondo voi, chi dovrebbe essere responsabile della sicurezza del software in uso: l'IT, lo sviluppatore, il fornitore o il business?
- Quanti ambienti distinti usate oggi per sviluppo, test e produzione?
- Quale sarebbe il primo software che dovreste aggiornare… ma che non osate toccare?


EDU TIPS
Q&A

Thank you!

OLIVIERO MAESTRELLI

R&D @smeup LAB

oliviero.maestrelli@smeup.com

EDU TIPS CYBERSECURITY