

Web server ed applicazioni web.

Principali problematiche di sicurezza, tipici errori implementativi

Webbit 2003 - 10 Maggio - Padova



<http://www.infosec.it>

info@infosec.it

Relatore: Igor Falcomatà - igor@infosec.it
IT Security Manager

<http://www.infosec.it/progetti.html>

Web security ed applicazioni web

audience: livello tecnico medio-basso

requisiti: nozioni base di ICT Security e networking

- **in questa presentazione tratteremo prevalentemente le problematiche di sicurezza legate ai web server ed alle applicazioni web**
- **con l'obbiettivo di fornire una panoramica/check list ed elencare alcune risorse utili per approfondimenti**
- **molti dei principi esposti sono validi per altre tipologie di applicazioni**
- **si presuppone una conoscenza dei principi base di sicurezza informatica e networking**

Web security ed applicazioni web

problematiche di sicurezza sempre più attuali.

- **spesso il web server è uno dei pochi servizi disponibili "pubblicamente"**
(quasi tutte le reti utilizzano firewall o simili)
- **utilizzato per veicolare contenuti e dati di importanza rilevante**
(banking/trading, b2b/b2c, news, intranet, messaging, ...)
- **grande complessità nel software**
(os, web/application server, CMS, database, ...)
- **grande complessità delle "personalizzazioni"**
(applicazioni sviluppate "in house" o proprietarie, ...)

Principali problematiche..

dalla resistenza di un singolo anello
dipende la sicurezza dell'intera catena

- **dati in transito**
- **dati sul sistema**
- **server**
- **altri server/sistemi coinvolti**
- **persone (utenti, amministratori, ...)**
- **web server e relativi moduli**
- **applicazioni web**

Sicurezza dei dati in transito

l'insicurezza intrinseca del protocollo TCP/IP e delle reti espone i dati in transito a diverse tipologie di attacco

- **sniffing (analisi del traffico)**
per catturare il traffico in transito e visualizzare i dati trasmessi (anche su reti switched)
- **hijacking (intromissione)**
per superare one-time-password e/o inserirsi in connessioni ESTABLISHED (è da notare che l'HTTP non si compone di un'unica connessione)
- **spoofing (impersonificazione)**
per superare autenticazioni "deboli" basate solamente sull'indirizzo IP o simile
- **man in the middle "avanzato" ("SSL sniffing")**
per analizzare il traffico di connessioni cifrate, falsificando i certificati forniti

Sicurezza dei dati in transito

spesso si valutano solamente le problematiche di analisi del traffico tra client/utente e server web...

- è opportuno considerare in fase di progetto **tutto il flusso di dati e relativi soggetti coinvolti**

application server, autenticazione, database, mainframe, mail, proxy, acceleratori/balancer/fw, dmz, lan, intranet, banche, magazzino, ordini, amministrazione, fornitori, partner, ...

... altrettanto spesso si ritiene che le problematiche siano relative solamente alle credenziali di accesso. Tutti i dati trasmessi in chiaro (in un punto qualsiasi del flusso) potrebbero risultare utili ai fini di un attacco.

SSL e cifra del traffico

"questo web server è sicuro perché utilizza SSL" (?!)

- **SSL serve a**

- proteggere i dati in transito (cifatura, integrità)
- garantire l'identità dei soggetti coinvolti
se verifichiamo l'autenticità dei certificati

- **ma ricordiamoci che SSL**

- non protegge i dati prima e dopo (sul client, sul server, ...)
- non protegge da compromissioni della struttura di PKI
- non è invulnerabile (errate implementazioni, ...)
- non è indecifrabile (per lo meno utilizzando "weak crypto")
- non garantisce la sicurezza delle reti e dei sistemi coinvolti

Sicurezza dei dati sul sistema

i dati presenti nella memoria del sistema possono essere acceduti e modificati

- **accesso diretto o via rete agli archivi**
utenti legittimi od illegittimi, superuser, acl, file/db temporanei, ...
- **accesso diretto o via rete attraverso applicazioni**
SQL ed altri database, acl sul database e sulle applicazioni, motori di ricerca, file/content serving, ...
- **accesso attraverso la memoria o tramite lib/call "hijacking"**
per accedere a password, chiavi crittografiche, dati, ...
- **Utenti non privilegiati potrebbero sfruttare lacune nella gestione di acl, permessi, privilegi, ...**
- **Utenti con credenziali di amministrazione hanno il controllo completo della macchina e dei dati**

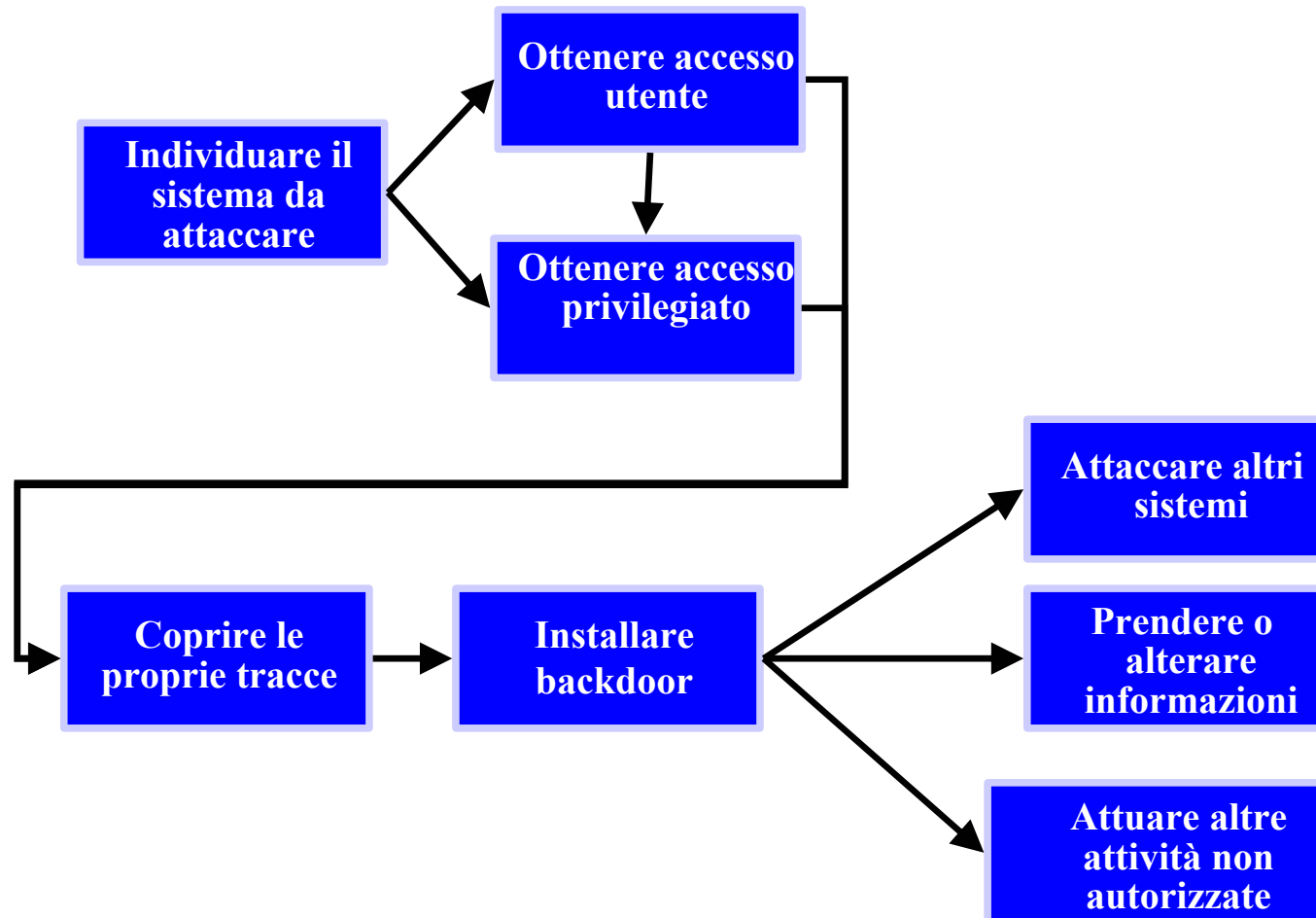
Sicurezza del sistema

vulnerabilità presenti in altre componenti/servizi/software possono compromettere la sicurezza dell'applicazione web

- **attacco attraverso servizi vulnerabili**
qualsiasi servizio disponibile da remoto potrebbe essere attaccato: web server e componenti, smtp/pop3/imap, servizi smb/dce, rpc/nfs, shell/remote management, SQL/database, p2p, dns, ...
- **privilege escalation e attacchi da locale**
utenti legittimi od illegittimi che possano accedere o modificare file ed archivi, eseguire codice, attacco attraverso "loopback", ...
- **altri attacchi**
 - vulnerabilità nello stack di networking
 - vulnerabilità nei client (resolver, browser, ...)
 - modifica di exe/archivi/dati/... su serv. remoti (fs, db, auth, ...)
 - vulnerabilità in qualsiasi applicazione che processi dati forniti, anche "indirettamente" e in minima parte, da fonte "untrusted"

Sicurezza del sistema

esempio di attacco



Web server ed applicazioni web – Problematiche di sicurezza - Pagina 10

© 2002-2003 Infosec srl - E' vietata la riproduzione di questo documento o sue parti in qualsiasi forma e su qualsiasi supporto.

Sicurezza del sistema

purtroppo è facile dimenticare qualcosa

- **"vulnerabilità"**

in questo caso intendiamo la possibilità di utilizzare un software/applicazione/componente per ottenere vantaggi in un attacco.

- **per esempio:**

errori di programmazione e/o implementazione, errori di configurazione, eccessiva liberalità nell'utilizzo o concessione di privilegi, utenti o credenziali di default o facilmente prevedibili, information leaking, traffico inviato o mantenuto "in chiaro", utilizzo senza validazione di dati "untrusted", autenticazione "debole", ...

In God we Trust

all others must submit an X.509 certificate. (C. Forsythe)

- **dati "untrusted"**

qualsiasi dato di cui non possiamo sapere con certezza in anticipo contenuto, sorgente, forma, ...

Ad esclusione dei dati che abbiamo creato noi e che non possono assolutamente essere stati modificati, sarebbe opportuno considerare "untrusted" qualsiasi cosa.

- **"con certezza" ed "assolutamente"?**

concetti piuttosto aleatori quando parliamo di sicurezza informatica...

Sicurezza dell'infrastruttura

vulnerabilità presenti in altri sistemi possono compromettere la sicurezza dell'applicazione web

- **altre componenti collegate all'applicazione web**
n-tier, database, auth. server, gateway wap/sms, test/backup, ...
- **infrastrutture di networking e sicurezza**
altre reti (anche non "pubbliche"), router, switch, balancer, firewall, ids, macchine del caffè', ...
- **sistemi con relazioni di fiducia**
stessa rete fisica, politiche di accesso privilegiate, userbase comuni, ...
- **fonti "indirette"**
mail/dns/fs, software distribution, update os/av, directory serv., ...

Persone

spesso l'anello debole della catena è rappresentato dalle persone, in particolar modo il personale non tecnico...

- **social engineering**
information leakage, abuso di buona fede, truffe, ...
- **disattenzioni, errori, mancanza di competenze**
nella progettazione, impostazione, manutenzione ed utilizzo di reti, software e sistemi...
- **worst practices**
password e credenziali di accesso banali, utilizzo di terminali pubblici/condivisi, cattiva protezione dei dati, condivisione delle credenziali, foglietti/post-it, ...
- **attacco a stazioni personali**
per accedere/modificare dati, credenziali, certificati, ...; per analizzare i dati in transito (sniffing) od immessi (snooping); per utilizzare privilegi di accesso altrui, ...

Web server, moduli, applicazioni..

come ogni software, sono soggetti ad errori di progettazione od implementazione che potrebbero renderli vulnerabili

- **cattiva validazione dell'input (e dell'output)**
buffer overflow & co., format bug, attacchi specifici per i linguaggi di programmazione/scripting utilizzati (in ogni componente)
- **design carente**
complessità intrinseca, numero sovrabbondante di moduli/componenti, eccessivi privilegi di esecuzione, mancanza di compartimentazione, autenticazioni "deboli", protocolli "deboli", backdoor(?), tmp race e carenze nella gestione di file/archivi/database, ...
- **cattiva configurazione**
impostazioni e configurazioni troppo permissive (spesso "by default"), mancata disabilitazione di moduli/componenti non utilizzati, "dimenticanze" od errori, mancanza di hardening, account/password di default, ...

Web server, moduli, applicazioni..

specifiche vulnerabilità, metodologie di attacco ed enumerazione, debolezze "strutturali"

- **web server**
Microsoft IIS, Apache HTTPD, Sun One, ...
- **application server, cms, etc.**
Lotus Domino, Oracle Application Server, IBM WebSphere, BEA WebLogic, Macromedia Coldfusion, Apache Jakarta, ...
- **linguaggi di programmazione**
ASP, PHP, Perl, shell script, C/C++, C#, Java, python, ...
- **database**
Oracle, IBM DB2, Microsoft SQL, Sybase, Informix, Interbase, MySql, ...
- **balancer, acceleratori, proxy & co**

Web server, moduli, applicazioni..

metodologie di attacco "generico"

- **enumerazione, information gathering**
fingerprinting, url, utenti, virtual host, archivi, protocolli, directory listing, engine di ricerca, information leakage, ...
- **bruteforcing, "black box testing", "fuzzy testing", ...**
protocolli, url, file/directory/archivi di test/backup & co, utenti, password, acl, sessioni, cookies, campi header, form, parametri CGI ed applicazioni, fqdn/vhost, ...
- **common techniques**
CGI scan, system call injection, character encoding, path traversal, SQL injection, cross site scripting & malicious code, ...

Web server, moduli, applicazioni..

enumerazione ed information gathering: obiettivi

- ottenere quante più informazioni possibili sulla struttura, il design, il funzionamento, la configurazione del sito/sistema, ...
- identificare software ed applicazioni utilizzate, moduli attivi, versioni, configurazioni, stato di aggiornamento e manutenzione, ...
- identificare utenti, amministratori ed altre persone coinvolte e loro abitudini, ...
- identificare form, pagine dinamiche, forum/messaging/intranet, sezioni "nascoste" o riservate, sezioni protette da password, ...

Web server, moduli, applicazioni..

**enumerazione ed information gathering: modalità
tools: ie/netscape, netcat, perl LWP, babelweb, wget, pavuk**

- web browsing: navigare nel sito ed accedere alle varie sezioni
- mirroring: utilizzare un programma "spider" per effettuare una copia/indicizzazione completa del sito
 - link interni
 - sezioni del sito
 - CGI e pagine dinamiche
 - mailto: (utenti?)
 - documenti da scaricare (informazioni su utenti, struttura di rete, mem garbage, password bruteforcing, ...)
 - link esterni
 - virtual host?
 - siti collegati?
 - partner/fornitori/consulenti/clienti?
 - host con particolari relazioni di fiducia?

Web server, moduli, applicazioni..

**enumerazione ed information gathering: modalità
tools: ie/netscape, netcat, perl LWP, babelweb, wget, pavuk**

- headers, cookies, formato degli url, messaggi d'errore, banner, trailer, sorgenti di pagine, script & co e relativi commenti: le informazioni fornite sono maggiori di quelle visualizzate attraverso un browser.
Esistono specifiche tecniche per identificare (fingerprinting) il software utilizzato, la configurazione, i moduli attivi, la versione ed ottenere altre informazioni utili ai fini di un attacco.
- metodi HTTP ed estensioni (HEAD, OPTIONS, DAV, ...)
- search engine esterni: attraverso i motori di ricerca è possibile identificare tutti i siti che "linkano" il sistema bersaglio (informazioni aggiuntive, relazioni di fiducia, partner/fornitori/consulenti, ...) e cercare informazioni sugli utenti (mailing list, newsgroup, siti personali, ...)
- motore di ricerca interno: potrebbe indicizzare sezioni "nascoste" del sito (o comunque non facilmente indicizzabili con uno spider), potrebbe indicizzare file/sezioni protette, backup, etc.

Web server ed applicazioni web – Problematiche di sicurezza - Pagina 20

Web server, moduli, applicazioni..

bruteforcing, "black box testing", "fuzzy testing": obiettivi

- identificare vulnerabilità non note (b0f, format bug, ...)
- identificare vhost/ip/fqdn
- identificare url/file/sezioni nascosti, sorgenti degli script, sezioni di test, private, di amministrazione, ...
- identificare CGI/applicazioni dinamiche con cattiva validazione dell'input
- forzare/bypassare/comprendere il funzionamento di CGI/applicazioni, sessioni, password, etc., provando ad immettere input mirato o pseudo-causale ed analizzando i responsi, gli errori, etc.
- identificare/forzare ACL (GET, POST, PUT, DELETE, "url based", ...)
- identificare/forzare credenziali di accesso (o almeno identificare utenti validi)

Web server, moduli, applicazioni..

**bruteforcing, "black box testing", "fuzzy testing": modalità
tools: b-e-f, spike, perl & shell scripting, netcat, wget**

- forzare i limiti, violare RFC, inserire caratteri particolari, inserire input/dati unexpected, casuali, etc. (url, campi dell'header, richieste, parametri, ...)
- utilizzare dizionari di parole, liste di nomi e loro variazioni per trovare password, utenti, vhost/fqdn, directory, file, CGI, parametri e quant'altro...
- partendo dai risultati dello spider e dalle altre informazioni raccolte, provare i vari metodi HTTP (HEAD, GET, PUT, POST, DELETE, MOVE ed estensioni varie specifiche) per forzare ACL, modificare dati, etc.
- partendo dai risultati dello spider e dalle altre informazioni raccolte, verificare la presenza di file di backup, temporanei, etc. (.bak, .old, prova_*, *~, ...)

Web server, moduli, applicazioni..

common techniques: CGI scan

- **obiettivi:**
individuare CGI/applicazioni dinamiche che abbiano vulnerabilità note
- **tools:**
whisker/libwhisker, nikto, babelweb, white hat arsenal, webinspect, vulnerability scanner specifici (DominoScan, OraScan, ...), vulnerability scanner generici (ISS, Nessus, GFI, WebTrend, Cisco, Typhon II, ...), etc.

Web server, moduli, applicazioni..

altre tecniche mirate a sovvertire il normale funzionamento, in particolar modo cattiva validazione di input...

- **obbiettivi:**

identificare comportamenti anomali, possibilità di sovvertire il normale funzionamento delle applicazioni, trovare i casi limite non previsti dagli sviluppatori

- **modalità:**

modificare qualsiasi parte/componente/richiesta inserendo input "fuori standard", inserire caratteri non previsti, sfruttare specifiche vulnerabilità. Al solito, in ogni parte: url e parametri, header, cookie, campi dei form, ...

- **tools:**

netcat, perl & moduli vari, shell scripting, spike & fuzzy tester, "security" proxy (achilles & co), ...

Strumenti utili [1/2]

- The Open Web Application Security Project

<http://www.owasp.org/>

- Guide to Building Secure Web Applications V 1.1.1, aa.vv.

http://sourceforge.net/project/showfiles.php?group_id=64424

- Secure Programming for Linux and Unix HOWTO, David A. Wheeler

<http://dwheeler.com/secure-programs/>

- Web & Application Security : validazione input & output, SQL Injection, XSS, problematiche specifiche legate a web/application server e linguaggi

<http://www.owasp.org/whitepapers/index.shtml>

- NGSSoftware / Research / Papers : metodologie di attacco a server IIS, Oracle AS, Lotus Domino; problematiche di SQL Injection

<http://www.nextgenss.com/research/papers.html>

- SPIDYNAMICS : SQL Injection e sicurezza delle applicazioni (web)

<http://www.spidynamics.com/whitepapers.html>

- Secure Programming, Oliver Friedrichs

<http://www.cli.di.unipi.it/~zoppi/docs/secprog.html>

Strumenti utili [2/2]

- **The World Wide Web Security FAQ, Lincoln D. Stein, John N. Stewart**
<http://www.w3.org/Security/Faq/www-security-faq.html>
- **Secure programming, Open Source Security Testing Methodology**
<http://www.ideahamster.org/projects.htm>
- **Make your software behave: CGI programming made secure (Gary McGraw, John Viega)**
<http://www-106.ibm.com/developerworks/library/secure-cgi/>
- **Mailing list: bugtraq, websecapp, pen-test, secprog, sectools, secpapers**
<http://online.securityfocus.com/archive>
- **Writing Secure Web Applications, Advosys Consulting**
<http://advosys.ca/papers/web-security.html>
- **Buffer overflow e dintorni, Nicola Bonelli**
<http://www.blackhats.it/code.ppt>
- **Programmazione sicura, Orlando Bassotto**
http://www.sikurezza.org/webbit02/Programmazione_sicura.ppt

Domande?

<Pubblicità Progresso>

<http://www.sikurezza.org/>
Italian Security Mailing List

</Pubblicità Progresso>



Web server ed applicazioni web.

Principali problematiche di sicurezza, tipici errori implementativi

Webbit 2003 - 10 Maggio - Padova



<http://www.infosec.it>

info@infosec.it

Relatore: Igor Falcomatà - igor@infosec.it
IT Security Manager

<http://www.infosec.it/progetti.html>