

Attacchi alle Applicazioni Web



<http://www.infosec.it>

info@infosec.it

Relatore: Matteo Falsetti

- **Applicazioni web:** *definizioni, scenari, rischi*
- **ICT Security e Web:** *scarsa consapevolezza*
- **Sviluppo di applicazioni web:** *errori comuni*
- **Attacchi al web:** *tecniche e contromisure*

Applicazioni Web (1/4)

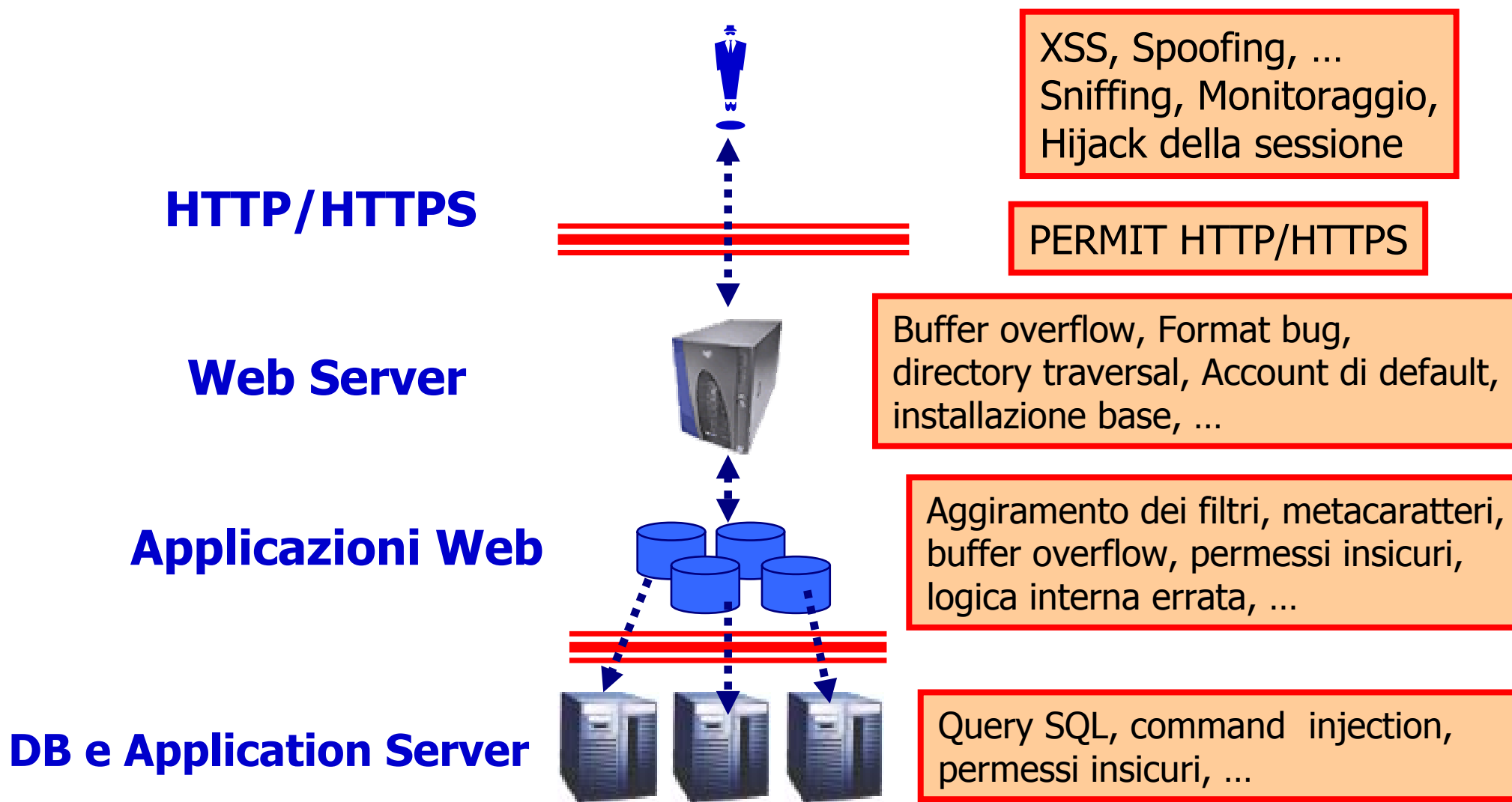
- ❑ Un'applicazione web non è altro che un software raggiungibile e utilizzabile mediante il protocollo **HTTP/HTTPS** ed un qualunque browser

- ❑ Se è raggiungibile alle porte **80/tcp** o **443/tcp** è con molta probabilità una applicazione web

Applicazioni Web (2/4)

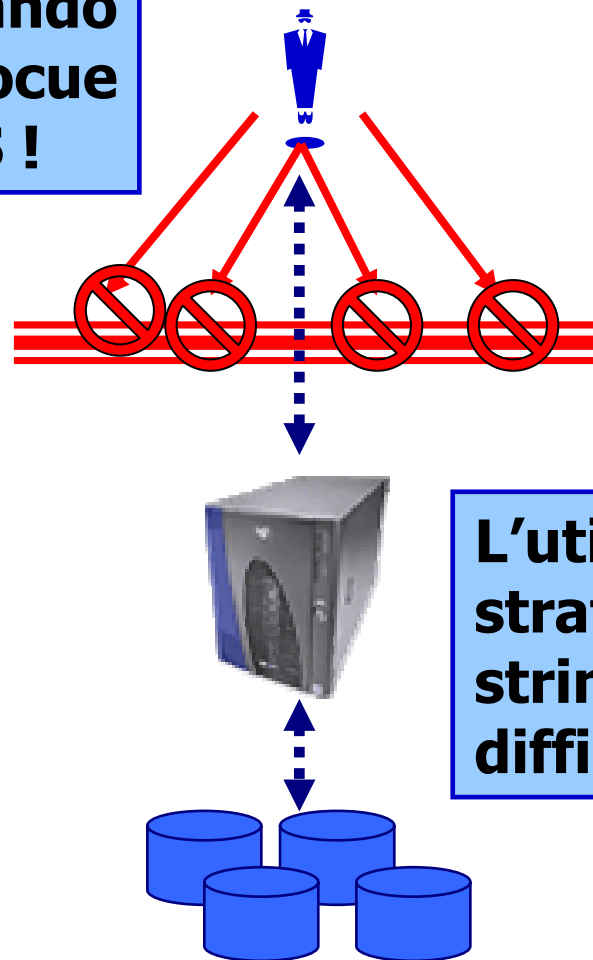
- Aste on-line, motori di ricerca, servizi web-mail, interrogazioni a database, forum e chat, home banking, gateway SMS, ...
- Negli ultimi dieci anni lo scenario del World Wide Web ha rivoluzionato i confini di Internet
- Gli strumenti di enforcing delle policy di sicurezza arrancano all'inseguimento del "layer applicativo"

Applicazioni Web (3/4)



Applicazioni Web (4/4)

Il firewall permette i servizi web, veicolando così l'attacco in innocue query HTTP e HTTPS !



L'utilizzo di SSL protegge lo strato di trasporto e cifra la stringa di attacco, rendendo difficile il compito degli IDS !

ICT Security e Web: *scarsa consapevolezza*

- ❑ È forse oggi il metodo più semplice di compromissione di un sistema e dei suoi utenti
- ❑ Le applicazioni web sono ubiquitarie
- ❑ Notevole complessità per i sistemi di Intrusion Detection
- ❑ Mancanza del supporto di screening di un firewall a livello applicativo
- ❑ Il sistema di cifratura privo di autenticazione non offre alcuna protezione
- ❑ Il livello di sviluppo delle applicazioni in ottica di sicurezza è ancora molto scarso

Sviluppo di applicazioni web: *errori comuni*

- ❑ Fidarsi dell'**input utente**
- ❑ Caratteri speciali non filtrati
- ❑ Output **HTML** non filtrato
- ❑ Sovrastima dei permessi necessari (**SUID**)
- ❑ Mancanza di autenticazione dell'utente remoto

Sviluppo di applicazioni web: *errori comuni*

Errato Input-Validation

- ❑ Fidarsi dei dati inviati dall'utente è la causa più importante delle vulnerabilità a livello applicativo
- ❑ È necessario identificare ogni parametro che possa contenere input utente

Sviluppo di applicazioni web: *errori comuni*

Caratteri Speciali

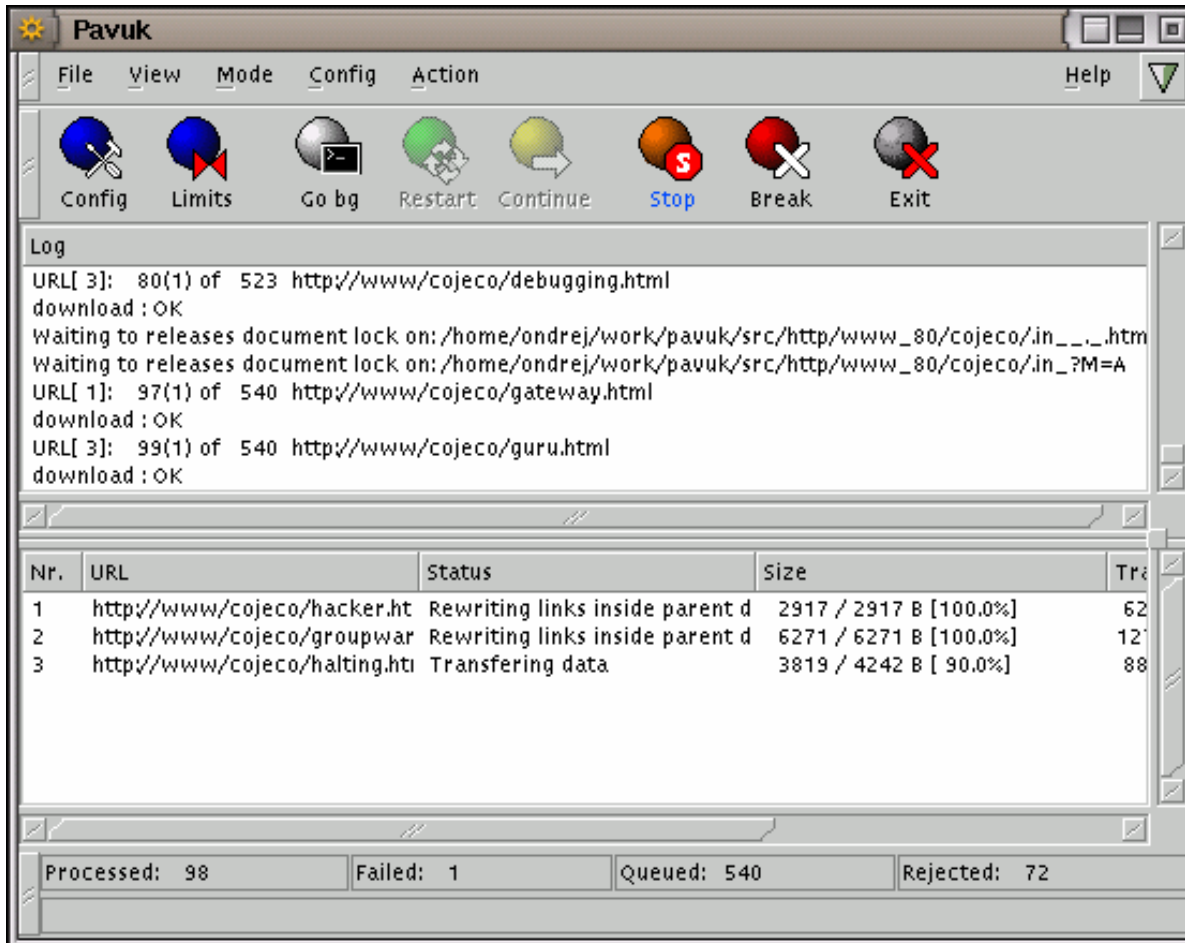
! @ \$ % ^ & * () - _ + ` ~ \ | [] { } ; : ' " ? / , . > <

- ✓ **Il mancato parsing di questi caratteri in ogni parametro può essere altamente pericoloso !**

```
http://troppo.comune.com/search.cgi?q=stringa&file=/html/search.db
```

```
http://troppo.comune.com/search.cgi?q=root&file=../../../../../../etc/passwd
```

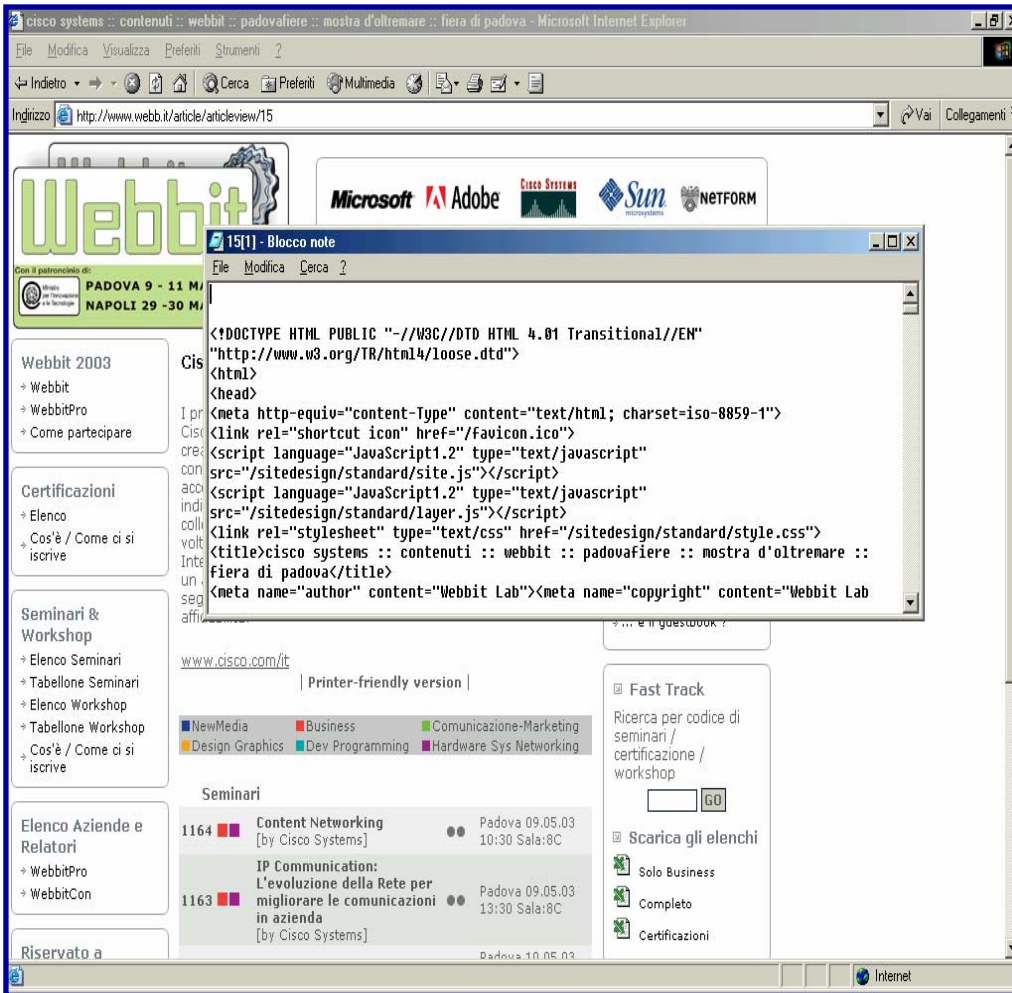
Attacchi al web



WEB SPIDERING

- ❑ Mappa del sito
- ❑ Documentazione di default
- ❑ Servizi nascosti
- ❑ CGI, campi di login, search engine
- ❑ Indirizzi e-mail

Attacchi al web



L'analisi del codice HTML identifica:

- Commenti originali degli sviluppatori
- Estensioni agli URL
- Cookies
- Linguaggi Client-Side
- Autenticazioni o "segreti"

OPTIONS * fingerprinting

```
PenTesting for Phun [2]
infosec@giringiro:~$ nc www.webb.it 80
OPTIONS * HTTP/1.1
Host: www.webb.it

HTTP/1.1 200 OK
Date: Fri, 09 May 2003 17:18:15 GMT
Server: Apache/1.3.23
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPAT
H, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE

punt!
infosec@giringiro:~$ nc www.microsoft.it 80
OPTIONS * HTTP/1.1
Host: www.microsoft.it

HTTP/1.1 302 Object moved
Server: Microsoft-IIS/5.0
Date: Fri, 09 May 2003 17:17:34 GMT
P3P: policyref="http://www.microsoft.com/w3c/p3p.xml" CP="ALL IND DSP COR ADM C
No CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR
UNI"
Location: http://www.microsoft.com/italy/%3CRejected-By-UrlScan%3E?~*
Content-Length: 180
Content-Type: text/html
Set-Cookie: ASPSESSIONID$SARAATRB=JBICHENCJLOMGOMCMNBAJIOO; path=/
Cache-control: private

<head><title>Object moved</title></head>
<body><h1>Object Moved</h1>This object may be found <a HREF="http://www.microso
t.com/italy/%3CRejected-By-UrlScan%3E?~*">here</a>.</body>
```

Possibili Tecniche di Attacco

- ❑ Manipolazione dei parametri di una applicazione
 - ❑ Valori non validi (type mismatch, out of range, ...)
 - ❑ Command injection
 - ❑ Uso esplicito di parametri "nascosti"

- ❑ Manipolazione dei cookie
 - ❑ Aggiramento delle procedure di autenticazione

- ❑ Ricerca di file sensibili
 - ❑ Backup di configurazioni, log, sorgenti di sviluppo, ...

Vulnerabilità Statiche

- ❑ Exploit conosciuti contro il server
 - ❑ CodeRed, Nimda, Apache Chunked, ...

- ❑ Enumerazione del "web tree"
 - ❑ /docs, /logs, /stats, /admin, /scripts, /old, ...

- ❑ Richieste specifiche alle applicazioni
 - ❑ CGI noti per "dir traversal", "command execution", ...

Esistono innumerevoli security scanner in grado di testare un server contro questi problemi. Ma le vulnerabilità maggiori e meno conosciute non sono di tipo statico !

Vulnerabilità Dinamiche

- ✓ Uno scanner automatizzato è in grado di controllare la presenza di **cgi-bin/printenv.pl** ma un membro del team di sviluppo ha spostato l'applicazione in **non_publico/sample-cgi/printenv.pl**
- ✓ Lo stesso scanner è sempre in grado di richiedere **cgi-bin/formmail.pl** ma non può conoscere applicazioni proprietarie come il nuovo e appena sviluppato **cgi-di-terze-parti/costoso.pl**

Una scansione automatizzata non può né dovrebbe sostituire una adeguata procedura di sviluppo e auditing delle applicazioni web !

Web Site Crawling ^(1/2)

Un attaccante ricostruirà la struttura del sito web bersaglio, analizzando ogni dato ottenuto alla ricerca di:

- ❑ il flusso logico applicativo
- ❑ directory nascoste ma referenziate nei motori di ricerca pubblici
- ❑ sezioni di pre-produzione delle pagine web
- ❑ vecchi documenti e applicazioni "dimenticate"
- ❑ materiale informativo inerente il bersaglio, procedure di sviluppo, i dipendenti, gli amministratori, essenzialmente ogni tipologia di dato "tralasciato" nelle pagine web

Web Site Crawling (2/2)

L'intera struttura ad albero del sito web deve essere controllata, al fine di rimuovere ogni vecchio dato, ogni informazione obsoleta, ma potenzialmente rilevante per un attaccante, ogni traccia di test e procedure di sviluppo !

Query Diretta delle Directory (1/2)

Ogni sito web di una certa complessità presenta un notevole numero di directory di livello sempre più profondo all'interno di una struttura ad albero. Un attaccante potrebbe scoprire **directory prive di documenti di default** ed ottenere il listato di tutti i file presenti, carpando importanti informazioni circa la natura del flusso applicativo del sito.

http://www.ops.it/d_2003/fatturato/premi/dettagli.html

http://www.ops.it/d_2003/fatturato/premi/ **I**

http://www.ops.it/d_2003/fatturato/ **II**

http://www.ops.it/d_2003/ **III**

Query Diretta delle Directory (2/2)

È importante assicurarsi che sia presente un documento di default in ogni directory o che sia alternativamente presente una modalità di redirect al documento principale. È altrettanto importante disabilitare nella configurazione del server la possibilità di fornire listati delle directory !

Hijack delle Sessioni

Un attaccante potrebbe predire o impadronirsi dell' **ID di sessione** o del valore di un **cookie** contenente i dati di autorizzazione, **aggirando** così i metodi di autenticazione preposti all'accesso di risorse protette.

In questo caso è importante eliminare ogni errore a livello di **sviluppo. È necessario effettuare un **audit del sorgente** per eliminare ogni generazione "**debole**" di identificativi e cookie di sessione !**

Documenti e Applicativi Nascosti (1/2)

All'interno di un documento HTML l'attaccante legge:

```
< ! – questo codice deriva da quell'obbrobrio di  
/test-scripts/buggato.asp – >
```

L'attaccante non solo scopre che **buggato.asp** sia ancora presente nel percorso **/test-scripts/**, ma anche che sia realmente prono a problematiche sfruttabili da remoto ...

Documenti e Applicativi Nascosti (2/2)

È necessario dotarsi di un ambiente di sviluppo e di beta test separato dall'ambiente di produzione ed inserire esclusivamente in questo commenti e versioni differenti delle applicazioni utilizzate !

Qualunque essere umano scrive qualche volta codice **insicuro**... ma questo non vuol dire che debba **comunicarlo a chiunque**...

Reversing di Applet Java

L'attaccante nota come una forma di **autenticazione** sia eseguita a **lato client** attraverso il download e l'esecuzione di un applicativo Java. Tutto ciò che occorre è la **decompilazione** e lo **studio** dell'applet per trovare eventuali credenziali o link ed informazioni utili ad aggirare le procedure di autenticazione

L'uso di un applet lato client **non è assolutamente sufficiente** per autenticare un utente remoto. Inoltre potrebbe essere utile eseguire alcune procedure di offuscamento delle classi per complicare il lavoro di reverse engineering.

Backup del sito

Spesso sono presenti **copie intere del sito web** in directory quali /old, /backup, **/solo_per_i_tuoi_occhi** ... Indipendentemente dal nome della directory, un attaccante potrebbe scoprire tali copie di backup ed analizzare la struttura del sito e delle sue applicazioni senza la normale limitazione di privilegi e policy di accesso.

Le copie di backup sono estremamente necessarie, ma **non** sui server di produzione ! Nel caso sia necessario mantenerle sugli stessi sistemi, è necessario **rimuoverli dal Web Tree** per evitarne l'accessibilità.

Parametri di Input (1/2)

Il problema peggiore della maggior parte delle applicazioni web è la **sconsiderata fiducia** riposta nell'**input utente**. Un attaccante analizzerà i parametri di ogni applicazione ed invierà loro valori palesemente **errati** o **arbitrari**

```
/search.cgi?query=trova$questo&banner=/text/legale.txt
```

```
/search.cgi?query=root&banner=/../../../etc/passwd
```

```
/search.cgi?query=|dir%20c:\
```

```
/search.cgi?query=";IFS='$';cat%20/etc/passwd
```

```
/search.cgi?query=AAAAAAAAAAAAAAAAAAAAAAAAAAAAA....AAAAAAAA
```

Parametri di Input (2/2)

Mai fidarsi dell'input fornito dall'utente ! La validazione deve essere effettuata sempre e comunque **server side** e non, come troppo spesso accade, operata mediante **Javascript client side** !

Cross Site Scripting (1/2)

- L'attaccante può forzare il server a fornire codice Javascript scelto arbitrariamente; questo sarà eseguito dal client nel contesto del sito web visitato
- Lo scenario tipico consiste nel mascherare procedure di autenticazione agli occhi del client, per sottrarre credenziali di accesso valide e cookie di sessione, oppure per forzare il client al download di codice malevolo
- È molto semplice per l'attaccante scoprire quali applicazioni e server siano vulnerabili:

```
/search.cgi?query=<script>alert('Vulnerabile al XSS')</script>
```

Cross Site Scripting (2/2)

Questo è un problema molto serio, in quanto la valutazione della pericolosità viene spostata sulle spalle dell'utente finale dell'applicazione, molto spesso non in grado di discernere tra richieste valide e malevoli, soprattutto nel caso queste siano sufficientemente precise da replicare il normale codice HTML che l'utente si aspetta...

L'applicazione deve **filtrare** ogni **codice HTML** dall'input utente. Alternativamente alcuni caratteri speciali come < e > dovrebbero essere tradotti mediante il tipico encoding HTML e non inviati al client come tali.

SQL Injection (1/2)

- Questo attacco, nonostante la sua semplicità, è presente in un gran numero di applicazioni web
- L'applicazione vulnerabile utilizza input fornito dall'utente all'interno di query SQL dirette ad un database, frequentemente protetto in un segmento di rete "interno", senza prima filtrare l'input stesso
- L'attaccante è così in grado di eseguire query SQL arbitrarie e compromettere il database

```
/login.asp?username=admin&password=h4K.J0qI
```

```
/login.asp?username='having 1=1--&password=.
```

```
/login.asp?username=.&password='or 1=1--
```

SQL Injection (2/2)

- ✓ Come per ogni altro problema di **validazione dell'input utente**, anche gli attacchi di SQL Injection possono essere bloccati semplicemente apponendo **adeguati filtri** sui caratteri forniti arbitrariamente.
- ✓ Nonostante il problema dell'input si presenti in svariate tipologie di attacco, nonostante sempre più applicazioni usino database SQL come back-end applicativo, questa tecnica di attacco è al momento **efficacemente usata** contro innumerevoli siti istituzionali per ottenere accesso non autorizzato a dati e risorse sensibili.

Applicazioni Web

Nuovo limite e confine dell'ICT Security

- Nel corso degli ultimi 10 anni il World Wide Web ha mutato considerevolmente non solo l'attitudine degli operatori e degli utenti in Internet, ma soprattutto le tipologie di attacco e information gathering dei sistemi informatici
- Al momento gli strumenti di gestione ed enforcing di policy di autorizzazione, autenticazione ed integrità delle risorse stentano a seguire il passo
- È necessario operare un controllo sistematico della logica interna di tali applicazioni per ovviare alla carenza di dispositivi validi di Intrusion Detection e di prevenzione attiva delle intrusioni

Attacchi alle Applicazioni Web



<http://www.infosec.it>

info@infosec.it

Domande ?