Hacker Highschool
SECURITY AWARENESS FOR TEENS

LEZIONE 10
SICUREZZA WEB
E PRIVACY

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

“License for Use” Information

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the Hacker Highschool web page at www.hackerhighschool.org/license.

The HHS Project is a learning tool and as with any learning tool, the instruction is the influence of the instructor and not the tool. ISECOM cannot accept responsibility for how any information herein is applied or abused.

The HHS Project is an open community effort and if you find value in this project, we do ask you support us through the purchase of a license, a donation, or sponsorship.

All works copyright ISECOM, 2004.

Informazioni sulla licenza d’uso

Le seguenti lezioni ed il materiale per gli esercizi (workbook) sono materiale di tipo "open" e liberamente disponibile al pubblico, secondo i termini e le condizioni di ISECOM. Per comprendere le nostre condizioni di utilizzo, o acquistare una licenza per utilizzi di tipo commerciale, vi invitiamo a visitare la sezione LICENSE di questo sito web.

Il Progetto HHS è uno strumento per apprendere e, come ogni strumento di questo tipo, la chiave formativa consiste nella capacità e nell'influenza dell'istruttore, e non nello strumento formativo. ISECOM non può accettare e/o farsi carico di responsabilità per il modo in cui le informazioni qui contenute possono essere utilizzate, applicate o abusate.

Il Progetto HHS rappresenta uno sforzo di una comunità aperta: se ritenete il nostro lavoro valido ed utile, vi chiediamo di supportarci attraverso l'acquisto di una licenza, una donazione o una sponsorizzazione al progetto.

Tutto il materiale è copyright ISECOM, 2004.

Indice

Contributors

Simon Biles

Pete Herzog, ISECOM

Bill Matthews

Hernán Marcelo Racciatti

Chris Ramirez

P. Shreekanth

Kim Truett , ISECOM

Marta Barceló, ISECOM

Dario Riquelme Zornow

Per la versione in lingua italiana:

Raoul Chiesa (ISECOM, Director of Communications, OSSTMM Key Contributor)

Doriano Azzena (centro CSAS del progetto Dschola IPSIA Castigliano - Asti)

Sophia Danesino (centro CSAS del progetto Dschola ITIS Peano – Torino)

Nadia Carpi (centro CSAS del progetto Dschola ITIS Peano – Torino)

Fabrizio Sensibile, OPST&OPSA Trainer (@ Mediaservice.net Srl, Torino – ISECOM Authorized Training Partner)

Claudio Prono (@ Mediaservice.net Srl, Torino – ISECOM Authorized Training Partner)

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

mediaservice.net
CORPORATE SECURITY & IMAGE

10.1 Fondamenti di sicurezza web

Quello che ognuno fa sul World Wide Web sono problemi suoi. O così si potrebbe pensare. Ma questo semplicemente non è vero. Tutto quello che si fa sul web è privato e anonimo come quando si esce di casa. In questo caso si potrebbe pensare che sono affari nostri e molti, inclusa ISECOM, potrebbero essere d'accordo. Tuttavia, considerate un investigatore privato che vi segue per la città, registrando tutto quello che avete visto e tutti quelli con cui avete parlato.

L'obiettivo di questa lezione è di insegnare a proteggersi sul web e, per far ciò, è necessario imparare dove si trovano i pericoli.

Il World Wide Web funziona in maniera semplice. Una volta che ci si connette ad Internet attraverso il proprio ISP, si apre un browser, si specifica un sito web e si ottiene il sito sullo schermo. Tuttavia la verità è nei dettagli. Come funziona realmente il web?

Una visita veloce al World Wide Web Consortium (W3C), quell'insieme di persone che si occupano di definire gli standard per il web, vi dirà tutto quello che volete sapere sul web. http://www.w3.org. Anche la storia del web: http://www.w3.org/History.html. Il problema è: le definizioni e gli standard vi insegneranno come essere sicuri? Apparentemente no. Le persone che vi vogliono male non seguono necessariamente gli standard.

10.1.1 Come funziona realmente il web

I passi seguiti durante una connessione ad Internet e successivamente al web sono molto dettagliati anche se sembrano semplici dal punto di vista dell'utente.

Cosa accade realmente quando volete semplicemente connettervi al sito web di ISECOM? Supponete di essere già connessi ad Internet, qui sono elencati i passi che avvengono in ordine:

1.Aprite il vostro browser

2.Inserite la URL (nome del sito web)

3.Il nome del sito web viene salvato nella cache storica sull'hard disk

4.Il vostro computer cerca il nome dell'indirizzo nel server DNS di default per trovare l'indirizzo IP

5.Il vostro computer si connette al server all'indirizzo IP fornito sulla porta di default 80 TCP se avete usato "HTTP://" o 443 TCP se avete usato "HTTPS://" prima del nome del server web (tra l'altro se avete usato HTTPS ci sono altri passi che riguardano l'utilizzo di certificati che non abbiamo seguito in questo esempio)

6.Il vostro computer richiede la pagina o la cartella che avete specificato, utilizzando come default se non specificate nulla spesso "index.htm" . Ma è il server che decide il default non il vostro browser.

7.Le pagine vengono memorizzate in una cache sul vostro hard disk. Anche se gli dite di memorizzare le informazioni in memoria (RAM), c'è una buona probabilità che finiscano da qualche parte sul disco o in un PAGEFILE o in uno SWAPFILE

8.Il browser quasi istantaneamente mostra cosa è stato memorizzato. Nuovamente c'è una differenza tra la "velocità percepita" e la "velocità reale" della vostra navigazione web che è la differenza tra la velocità del download (reale) e quella con cui il browser e la scheda

grafica può generare la pagina e la grafica e mostrarla (percepita). Solo perchè non la vedete questo non vuol dire che che non si trovi già nella cache del vostro browser.

La storia del World Wide Web (diremo solo "web" da ora in poi) ebbe inzio al CERN[1] nel 1989. Fu ideato da Tim Berners-Lee e Robert Cailliau che costruirono un sistema elementare basato su ipertesti per scambiare informazioni. Negli anni successivi Tim Berners-Lee continuò a sviluppare il sistema fino a che nel 1993 il CERN annunciò che chiunque poteva usare il web ed il web che ora vediamo esplose sulla scena.

---

[1] Centre Européen pour la Recherche Nucléaire(European Centre for Nuclear Research)

Il Web è un concetto basato su un client e un server, con un client, come Internet Explorer, Firefox, Mozilla, Opera, Netscape e altri, che si connette a un server web come IIS e Apache, che a sua volta fornisce contenuti nella forma di pagine HTML [1] . Molte compagnie, organizzazioni e individui hanno collezioni di pagine web ospitate su server e distribuiscono una grande quantità di informazioni a tutto il mondo.

Perchè ci dovremmo preoccupare della sicurezza del web? I server web spesso sono equivalenti alla finestra del negozio di una ditta. E' un posto dove si pubblicizzano e esibiscono informazioni, ma questo si suppone essere sotto il vostro controllo. Ciò che non volete è lasciare la finestra aperta così che chiunque possa raggiungerla e prendere quello che vuole gratuitamente e idealmente volete essere sicuri che se qualcuno lancia un mattone, la finestra non si rompa! Sfortunatamente i server web sono programmi complessi e come tali hanno un'alta probabilità di contenere dei bachi e questi ultimi vengono scoperti ed utilizzati da membri meno scrupolosi della società per ottenere l'accesso a dati che non dovrebbe essere visti.

E anche il contrario è altrettanto vero. Ci sono anche rischi legati al lato client del vostro browser. C'è un numero di vulnerabilità che sono state scoperte nello scorso anno che consentono ad un sito web di compromettere la sicurezza di una macchina client che si connette a loro.

---

[1] Hyper Text Markup Language

10.1.2 Sferragliare con le serrature

Le pagine HTML Standard vengono trasferite utilizzando HTTP[1], questo protocollo basato su TCP è puro testo e questo significa che è possibile effettuare facilmente connessioni ad un server utilizzando strumenti quali "telnet" o "netcat". Possiamo usare questa possibilità per ottenere una grande quantità di informazioni su quale software è in esecuzione su un server specifico. Ad esempio:

[1]Hyper Text Transfer Protocol

```
simon@exceat:~> netcat www.computersecurityonline.com 80
HEAD / HTTP/1.0
```

HTTP/1.1 200 OK
Date: Fri, 07 Jan 2005 10:24:30 GMT
Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
Last-Modified: Mon, 27 Sep 2004 13:17:54 GMT
ETag: "1f81d-32a-41581302"
Accept-Ranges: bytes
Content-Length: 810
Connection: close
Content-Type: text/html

Inserendo "HEAD / HTTP/1.0" seguito dal tasto "Return" premuto due volte, si possono ottenere tutte le informazioni precedenti sul server HTTP. Ogni versione e tipo di server HTTP fornirà informazioni differenti a questa richiesta – un server IIS restituirà:

```
simon@exceat:~> netcat www.microsoft.com 80
HEAD / HTTP/1.0
```

HTTP/1.1 200 OK
Connection: close
Date: Fri, 07 Jan 2005 11:00:45 GMT
Server: Microsoft-IIS/6.0
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"
X-Powered-By: ASP.NET
X-AspNet-Version: 1.1.4322
Cache-Control: public, max-age=9057
Expires: Fri, 07 Jan 2005 13:31:43 GMT
Last-Modified: Fri, 07 Jan 2005 10:45:03 GMT
Content-Type: text/html
Content-Length: 12934

Si possono ottenere ulteriori informazioni utilizzando la clausola "OPTIONS" nella richiesta HTTP come segue:

```
simon@exceat:~> netcat www.computersecurityonline.com 80
OPTIONS / HTTP/1.0
```

HTTP/1.1 200 OK
Date: Fri, 07 Jan 2005 10:32:38 GMT
Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
Connection: close

Questo vi darà tutti i comandi HTTP consentiti a cui il server risponderà.

Fare tutto questo a mano è piuttosto noioso, e confrontare manualmente tali informazioni con un database contenente le vulnerabilità e firme conosciute è più di quando si vorrebbe fare. Fortunatamente per noi, alcune persone molto intraprendenti hanno sviluppato una soluzione automatizzata chiamata "nikto".

"Nikto" è uno script Perl script che effettua vari test automaticamente! Le opzioni sono le seguenti:

-Cgidirs+ effettua una scansione di queste dir CGI: 'none', 'all', o un valore come '/cgi/'

-cookies visualizza i cookie trovati

-evasion tecnica di evasione ids (1-9, si veda oltre)

-findonly trova solo le poerte http(s), non effettua una scansione completa

-Format salva file (-o) Formato: htm, csv or txt (presunto)

-generic forza una scansione completa (generica)

-host+ host obiettivo

-id+ autenticazione da utilizzare sull'host, il formato è userid:password

-mutate+ modifica i controlli (si veda oltre)
-nolookup non effettua la ricerca del nome

-output+ scrive l'output in questo file

-port+ port a da usare (default 80)

-root+ preporre il valore che assumerà la funzione di root directory in tutte le richieste, il formato è /directory

-ssl forza la modalità ssl sulla port

-timeout timeout (default 10 secondi)

-useproxy usa il proxy specificato nel config.txt

-Version visualizza i plugin e le versioni dei database

-vhost+ virtual host (per l'header dell'Host)

(+ significa che è richiesto un valore)

Le seguenti opzioni non possono essere abbreviate:

-debug modalità debug
-dbcheck controllo sintassi su scan_database.db e user_scan_database.db
-update aggiorna i database e i plugin da cirt.net
-verbose modalità verbosa

IDS Tecniche di evasione:

1 Codifica Casuale URI (non-UTF8)
2 Directory di riferimento (/./)
3 Termine prematuro URL
4 Preporre lunghe stringhe a caso
5 Parametro falso

6 TAB come spaziatura richiesta
7 Esegue richieste alternando in modo casuale maiuscola/minuscola
8 Utilizzo del separatore di directory Windows (\)
9 Intreccio di sessioni

Tecniche di mutazione:

1 Test di tutti i file con tutte le directory radice
2 Indovinare i nomi di file di password
3 Enumerare i nomi utenti tramite Apache (richieste /~user)
4 Enumerare i nomi utenti via cgiwrap (richiesta /cgi-bin/cgiwrap/~user)

"Nikto" è piuttosto esauriente nelle sue relazioni come potete vedere dalla seguente scansione:

```
exceat:/# ./nikto.pl -host www.computersecurityonline.com

---------------------------------------------------------------------------
- Nikto 1.34/1.29     -     www.cirt.net
+ Target IP:       217.30.114.2
+ Target Hostname: www.computersecurityonline.com
+ Target Port:     80
+ Start Time:      Fri Jan  7 12:23:56 2005
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/1.3.27 Ben-SSL/1.48 (Unix) PHP/4.2.3
- Server did not understand HTTP 1.1, switching to HTTP 1.0
+ Server does not respond with '404' for error messages (uses '400').
+     This may increase false-positives.
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND,
PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
+ HTTP method 'PUT' method may allow clients to save files on the web server.
+ HTTP method 'CONNECT' may allow server to proxy client requests.
+ HTTP method 'DELETE' may allow clients to remove files on the web server.
+ HTTP method 'PROPFIND' may indicate DAV/WebDAV is installed. This may be used to get
directory listings if indexing is allowed but a default page exists.
+ HTTP method 'PROPPATCH' may indicate DAV/WebDAV is installed.
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled.
+ Apache/1.3.27 appears to be outdated (current is at least Apache/2.0.50). Apache 1.3.31 is
still maintained and considered secure.
+ Ben-SSL/1.48 appears to be outdated (current is at least 1.55)
+ PHP/4.2.3 appears to be outdated (current is at least 5.0.1)
+ PHP/4.2.3 - PHP below 4.3.3 may allow local attackers to safe mode and gain access to
unauthorized files. BID-8203.
+ Apache/1.3.27 - Windows and OS/2 version vulnerable to remote exploit. CAN-2003-0460
+ Apache/1.3.27 - Apache 1.3 below 1.3.29 are vulnerable to overflows in mod_rewrite and
mod_cgi. CAN-2003-0542.
+ /~root - Enumeration of users is possible by requesting ~username (responds with Forbidden
for real users, not found for non-existent users) (GET).
+ /icons/ - Directory indexing is enabled, it should only be enabled for specific directories
(if required). If indexing is not used all, the /icons directory should be removed. (GET)
+ / - TRACE option appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACE)
+ / - TRACK option ('TRACE' alias) appears to allow XSS or credential theft. See
http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf for details (TRACK)
+ /CVS/Entries - CVS Entries file may contain directory listing information. (GET)
+ /images/ - index of image directory available (GET)
+ /manual/ - Web server manual? tsk tsk. (GET)
+ /cgi-bin/cgiwrap - Some versions of cgiwrap allow anyone to execute commands remotely. (GET)
+ /cgi-bin/cgiwrap/~adm - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~bin - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
```

```
+ /cgi-bin/cgiwrap/~daemon - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~lp - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~root - cgiwrap can be used to enumerate user accounts. Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /cgi-bin/cgiwrap/~xxxxx - Based on error message, cgiwrap can likely be used to find valid
user accounts. Recompile cgiwrap with the '--with-quiet-errors' option to stop user
enumeration. (GET)
+ /cgi-bin/cgiwrap/~root - cgiwrap can be used to enumerate user accounts.  Recompile cgiwrap
with the '--with-quiet-errors' option to stop user enumeration. (GET)
+ /css - Redirects to http://www.computer-security-online.com/css/ , This might be
interesting...
+ 2449 items checked - 15 item(s) found on remote host(s)
+ End Time:        Fri Jan  7 12:25:36 2005 (100 seconds)
-------------------------------------------------------------------------
•  1 host(s) tested
```

Utilizzando altre opzioni potete impostare Nikto per fare esattamente quello che dovete ottenere incluse le modalità nascosto, mutazione e rilevazione dei cookies.

10.1.3 Uno sguardo a Tinted Windows -SSL

Non ci volle molto per capire che l'HTTP in chiaro non era il massimo per la sicurezza. Così la modifica successiva fu applicarvi la crittografia. Il risultato fu SSL[1] che utilizza un sistema di crittografia con una chiave pubblica ragionevolmente sicura da 40 o 128 bit. L'utilizzo di una chiave da 40 bit è meno sicuro di una a 128 e, con un hardware specializzato, un attacco di forza bruta può essere eseguito nell'ordine dei minuti, mentre lo stesso attacco con quella da 128 durerebbe più dell'età dell'universo. Tuttavia ci sono tecniche di attacco più complesse note come attacco di un testo cifrato di cui sono noti alcuni fattori – questo prevede il calcolo della chiave di crittografia dall'analisi di un grande numero di messaggi (più di un milione) per dedurre la chiave. In ogni caso non sarete molto veloci nel cercare di decifrare una crittografia a 128 bit – quindi cosa possiamo imparare sui server SSL HTTP? Abbastanza. Poiché SSL semplicemente crittografa il traffico standard HTTP, se impostiamo un tunnel SSL, possiamo interrogare un server come abbiamo fatto nella sezione 1.1. La creazione di un tunnel SSL è abbastanza immediata e esiste una utility detta "stunnel" apposta per tale scopo. Inserite in un file chiamato stunnel.conf, (sostitutendo ssl.enabled.host con il nome del server SSL a cui volete connettervi):

```
client=yes
verify=0
[psuedo-https]
accept  = 80
connect = ssl.enabled.host:443
TIMEOUTclose = 0
```

Stunnel mapperà la porta locale 80 con la porta remota 443 ed emetterà testo in chiaro in modo che vi possiate connettere utilizzando uno qualsiasi dei metodi presentati prima:

---

[1]Secure Sockets Layer

simon@exceat:~> netcat 127.0.0.1 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Fri, 07 Jan 2005 10:32:38 GMT
Content-type: text/html
Last-modified: Fri, 07 Jan 2005 05:32:38 GMT
Content-length: 5437
Accept-ranges: bytes
Connection: close

10.1.4 Avere qualcun altro che lavora per te – Proxy

I proxy si trovano a metà di una transazione HTTP. Il client fa la richiesta al proxy, il proxy effettua la richiesta al server, il server risponde al proxy e finalmente il proxy restituisce la risposta al client completando la transazione. I server proxy sono vulnerabili agli attacchi e sono anche un trampolino di lancio per attacchi ad altri server. Tuttavia possono aumentare la sicurezza filtrando le connessioni, da e verso i server.

10.2 Vulnerabilità del Web

La semplicità di dare qualcosa a qualcuno che lo chiede è resa più complessa quando si è nel business delle vendite. I siti web che vendono, compagnie che vedono prodotti, i bloggers che vendono idee e personalità o giornali che vendono notizie, richiedono più di un testo HTML e immagini. Pagine web dinamiche che aiutano a decidere di cosa far richiesta, mostrano alternative, raccomandano altre opzioni, cercano di vendere altri prodotti aggiuntivi, e vi danno solo quello che per cui avete pagato richiedono un software complesso. Quando diciamo addio ai siti web e benvenuto a applicazioni web ci troviamo in un nuovo mondo di problemi di sicurezza.

10.2.1 Linguaggi di scripting

Molti linguaggi di scripting sono stati usati per sviluppare applicazioni che consentissero alle aziende di proporre i propri prodotti o servizi al web. Nonostante ciò sia ottimo per la proliferazione delle opportunità di lavoro,, crea anche nuove strade di attacco per gli hacker. Il maggior numero di vulnerabilità delle applicazioni web non sono nei bachi del linguaggio scelto, ma nei metodi e procedure utilizzate per sviluppare applicazioni web così come nella modalità di configurazione del server web. Ad esempio, se un modulo richiede un codice postale e l'utente inserisce “abcde”, l'applicazione può fallire se lo sviluppatore non verifica correttamente i dati provenienti dal modulo. Vari linguaggi possono essere utilizzati per creare applicazioni web, inclusi CGI, PHP e ASP.

Common Gateway Interface (CGI): Whatis.com definisce una CGI come "Un mezzo standard con cui un server web passa la richiesta di un utente web ad un programma applicativo e per ricevere indietro i dati all'utente." CGI fa parte del protocollo web Hypertext Transfer Protocol (HTTP). Vari linguaggi possono essere usati per facilitare il programma applicativo che riceve e elabora i dati dell'utente. Le applicazioni CGI più popolari sono: C, C++, Java e PERL.

PHP – Hypertext PreProcessor (PHP): PHP è un linguaggio di scripting lato server open-source dove lo script è inserito in una pagina web insieme all'HTML. Prima che una pagina sia inviata ad un utente, il server web chiede al PHP di interpretare e effettuare le operazioni inserite nello script PHP. Laddove HTML mostra contenuti statici, il PHP consente allo sviluppatore di costruire pagine che presentano all'utente contenuti dinamici e basati sull'input dell'utente. Alle pagine HTML che contengono script PHP viene generalmente dato un nome file con il suffisso ".php".

Active Server Pages (ASP): le pagine web che hanno un'estensione .asp Active server pages (ASP) sono pagine web create dinamicamente a partire da un database. Utilizzano scripting ActiveX – generalmente VB Script o codice Jscript. Quando un browser richiede una ASP, il server Web genera una pagina contenente codice HTML e immediatamente lo invia indietro al browser – in questo modo consente agli utenti web di vedere i dati in tempo reale, ma sono più vulnerabili ai problemi di sicurezza.

10.2.2 Problemi Comuni alle Applicazioni Web

Le applicazioni Web non hanno necessariamente tipi di problemi specifici, ma usano termini specifici per identificare i problemi tipici del web. All'aumentare dei test sulle applicazioni web sono aumentati anche quelli relativi alla sicurezza e ne è scaturita una classificazione specifica per le applicazioni web. I problemi comuni che si riscontrano sulle applicazioni web sono stati classificati in accordo al OSSTMM Risk Assessment Values (http://www.isecom.org/securitymetrics.shtml), una modalità specifica di misurare la sicurezza in base a come influenza il funzionamento delle cose.

RAV	*Cosa significa*	*Esempi Web*
Autenticazione	These are the identification and authorization mechanisms used to be certain that the person or computer using the web application is the correct person to be using it.Sono i meccanismi di identificazione e autorizzazione usati per verificare che una persona o un computer che utilizzano un'applicazione web siano autorizzati a farlo.	Ogni volta che si accede ad una pagina web che ha i vostri dati personali vi state autenticando. autenticazione spesso significa semplicemente specificare una login e una password. Qualche volta comporta specificare un numero identificativo o anche semplicemente provenire da un indirizzo IP accettato (lista-accettata)

RAV	Cosa significa	Esempi Web
Non ripudio	Un record che prova che i dati inviati a o ricevuti dall'applicazione web siano realmente stati inviati e dove.	Nonostante voi possiate non vederlo la maggior parte delle applicazioni web tengono traccia degli acquisti che sono stati effettuati da un particolare indirizzo IP utilizzando un particolare browser su un particolare sistema operativo come un record. E' possibile, quindi, che risulti molto probabile che sia stato qualcuno dal vostro computer ad aver effettuato un particolare acquisto. Senza un'autenticazione specifica, tuttavia, non è possibile determinare al 100% che siate stati proprio voi.
Segretezza	Un modo per assicurare che la comunicazione tramite applicazioni web non sia ascoltata da altre persone.	La parte HTTPS di interazione con una applicazione web fornisce una discreta segretezza. Rende il vostro traffico sul web difficilmente leggibile dal pubblico.
Privacy	Un modo per assicurare che la modalità con cui contattate e comunicate con un'applicazione web non possa essere pre-determinato da altre persone.	Nonostante sia molto raro, non è inimmaginabile che un'applicazione web che contiene informazioni molto private non sia nemmeno identificabile, a meno che non proveniate dal posto corretto e non conosciate la combinazione segreta per accedere all'applicazione web. Un modo è quello di dover cliccare su un'immagine in 5 posti diversi in un ordine prestabilito per ottenere la schermata di login. Un'altra maniera è chiamata *bussare alla porta* (port-knocking) e significa che il server richiede una sequenza di interazioni specifica prima di aprire una porta, come la porta HTTP, all'utente.
Identificazione	Queste sono le modalità per assicurare che l'applicazione web abbia protezione legale o, per lo meno, possa essere finanziariamente protetta da assicurazione.	Alcuni siti web scrivono chiaramente sulla schermata di accesso che è riservato a persone autorizzate. Se qualcuno ruba una login e una password o anche vi accede forzatamente, l'attaccante, se preso, non può affermare di non sapere che era privato.
Integrità	Questo è un record di validità della comunicazione con l'applicazione web per assicurare che ciò che è stato spedito e successivamente ricevuto dall'altro sia la stessa cosa e che, se alterata, sia	Alcune applicazioni web forniscono una "HASH" con i file che devono essere scaricati. Questa HASH è un numero generato da un file specifico. Quando si scarica il file è possibile confrontare l'HASH ricalcolato dal file con quello

RAV	Cosa significa	Esempi Web
	l'applicazione web che l'utente abbiano una traccia della modifica.	inviato. Questo assicura che alcuni attaccanti non vi ingannino con un file diverso o sostituendolo o con un altro o tramite raggiri, come in un Cross Site Scripting.
Sicurezza	Questo è il modo con cui si protegge un'applicazione web dai propri dispositivi di sicurezza. Se la sicurezza fallisce, è necessario essere sicuri che non alteri le operazioni dell'intera applicazione web.	E' possibile che un'applicazione utilizzi un demone che possa ri-inizializzarla o anche prevenire un attacco che causi il crash di qualunque sua parte presentandosi solo virtualmente. Potete anche trovare scenari in cui un'applicazione web utilizza un meccanismo di rilevazione intrusioni che ferma "attacchi" bloccando l'attaccante in base al suo indirizzo IP. In questo caso non possiamo affermare che ci sia sicurezza se il dispositivo di sicurezza non è configurato per evitare che un attaccante effettui spoofing sulle risorse dell'applicazione web e faccia sì che questa difesa blocchi traffico importante. Invece è considerato o una errata configurazione della difesa o in certi casi una debolezza della progettazione. Non confondete una difesa fatta poveramente o "accidentale", con una perdita di controllo progettata.
Usabilità	Un modo per evitare che l'utente debba prendere decisioni di sicurezza interagendo con l'applicazione web. Questo significa che la vera sicurezza è interna e che l'utente non deve scegliere quale meccanismo di sicurezza debba essere attivato o tolto.	When a web app requires use of HTTP over SSL (HTTPS) then we can say that it is using Usability as part of security. However, if it lets you choose to interact with it less securely, for example, to send your credit card number by insecure e-mail rather than post it via a form by way of HTTPS, then it is NOT exercising Usabilty.Quando un'applicazione web richiede l'uso di HTTP su SSL (HTTPS) possiamo dire che sta utilizzando *usabilità* come parte della sicurezza. Tuttavia, se vi si consente di scegliere se interagire con meno sicurezza, ad esempio, inviando il vostro numero di carta di credito tramite e-mail insicura piuttosto che inviarlo tramite un modulo che usa HTTPS allora NON sta praticando *usabilità*.
Continuià	Questo riguarda come evitare che un servizio basato su	Spesso un'applicazione web che riceve una grande quantità di traffico ha un

RAV	Cosa significa	Esempi Web
	un'applicazione web fallisca indipendentemente dal problema o dal disastro accaduto.	*reverse proxy* che direziona il traffico a uno dei molti server web *mirror*. In questo modo, se uno di essi cade il servizio non viene interrotto. Un altro esempio è un'applicazione web che effettua caching del proprio sito web su molti server in Internet in modo tale che quando ne visitate uno non vi trovate sul sito web originario. Se una cache cade o viene corrotta il traffico viene ridirezionato ad un'altra cache o al sito originario.
Allarme	Una notifica, o immediata o posticipata, che riguarda un problema con un uno di questi meccanismi.	Una forma base di allarme è il file di log generato dal server web. La cosa negativa di un allarme è che potete scegliere di ignorarlo. Questo è particolarmente vero se si verifica sempre (pensate alla storia del ragazzo che gridava "al lupo!"). O nel caso di un file di log può non allertare del tutto. Un allarme è utile in base al vostro tempo di reazione.

Esercizi:

1. Aprite google e digitate: "inurl:search.asp" o "inurl:search.php". Con uno qualunque dei siti web che compaiono cercate di digitare nel campo di ricerca **<script>alert ("hello")</script>**. Cosa accade? Fate la prova su più siti.

2. In google, digitate "inurl:login.asp" o "inurl:login.php". Con uno qualunque dei siti web che compaiono cercate di digitare i caratteri speciali (@#$^&) sia per lo username che per la password. Cosa accade? Fate la prova su più siti.

3. Conoscendo i tipi di meccanismi di sicurezza che può avere un'applicazione web, aprite il vostro sito web interattivo preferito e cercate di identificare se ha meccanismi di sicurezza che sono conformi a uno qualunque delle classificazioni RAV.

4. Le vulnerabilità discusse comunemente sono Cross Site Scripting (XSS) e *SQL injection*. Cosa sono e come vengono utilizzate da un attaccante per rubare dati e informazioni da un'applicazione web?

10.2.3 Linee guida per costruire applicazioni web sicure

Ci sono molte opinioni su come costruire un'applicazione in sicurezza e la maggior parte dei dettagli dipendono dalla logica del programmatore e dalla sua abilità con un linguaggio di programmazione. Esistono comunque anche delle linee guida, le seguenti provengono dai materiali disponibili da OSSTMM (http://www.osstmm.org).

1. Assicurare sicurezza non richiede che l'utente prenda decisioni
2. Richiedere una giustificazione commerciale per tutti gli input e output delle applicazioni
3. Mettere in quarantena e validare tutti gli input che includono contenuti dell'applicazione
4. Limitare la buona fede (al sistema e agli utenti)
5. Crittografare i dati
6. Effettuare l'hash dei componenti
7. Accertare che tutte le interazioni avvengano lato server
8. Strutturate a livelli la sicurezza
9. E' Meglio mostrare solo il servizio non le sue componenti
10. Scatenare allarmi
11. La consapevolezza della sicurezza è richiesta per gli utenti e helpdesks.

Esercizi:

1. Fornite esempi per ciascuna delle 3 linee guida precedenti

2. Specificate 3 tipi di tecnologie che si possono applicare ad un'applicazione web come allarme.

10.3 Fondamenti HTML – Una breve introduzione

HTML è un insieme di istruzioni che spiegano come le informazioni debbano essere presentate da un server web (Apache, Internet Information Server) a un browser (Firefox, Opera). E' il cuore del World Wide Web.

HTML può fare molto di più che semplicemente visualizzare dati su una pagina web. Può anche fornire moduli di acquisizione dati, dove i dati vengono inseriti per esere elaborati da linguaggi a livello più alto (Perl, PHP, etc). In un ambiente di lavoro l'HTML ha la massima utilità, ma per un hacker è vulnerabile al massimo.

10.3.1 Leggere HTML

HTML è composto da una serie di tag o markup. Ad ogni tag di apertura, <h1> ad esempio, deve corrisponderne uno di chiusura, </h1>. Questo specifica al browser di finire la formattazione specificata dal tag precedente. Tag di apertura e chiusura sono parti di un HTML ben strutturato.

Consideriamo, ad esempio, il codice:

```
<html>

<head><title>Ciao mondo</title></head>
<body>
<h1>Ciao mondo!</h1>
</body>
</html>
```

Figure 1: HTML Code

Stiamo dicendo al browser che questo è un documento HTML con il tag <html> e che avrà il titolo 'Ciao mondo' con il tag <title>. Il tag <body> indica al browser "qui è dove si trovano le informazioni che dovranno essere visualizzate." Infine il tag <h1> indica al browser di visualizzare le informazioni con lo stile "Heading 1". I tag preceduti con un /' sono semplicemente tag di chiusura, che indicano al broswer di terminare la visualizzazione dei contenuti descritti dal tag di apertura.

Esercizio 1: Incollate il codice di figura 1 in un file di testo chiamato hello.html. Aprite quel file in un browser a vostra scelta e vedrete qualcosa simile a questo:

Ciao mondo!

10.3.2 Visualizzare l'HTML come sorgente

Tutti i moderni browser prevedono un modo per visualizzare il codice HTML che ha generato la pagina HTML mostrata. Nella maggior parte dei casi è l'opzione "visualizza sorgente" nel menù "visualizza" del browser.

Esercizio 2: Selezionate Visualizza --> Visualizza sorgente nel vostro browser mentre navigate nella vostra pagina web preferita.

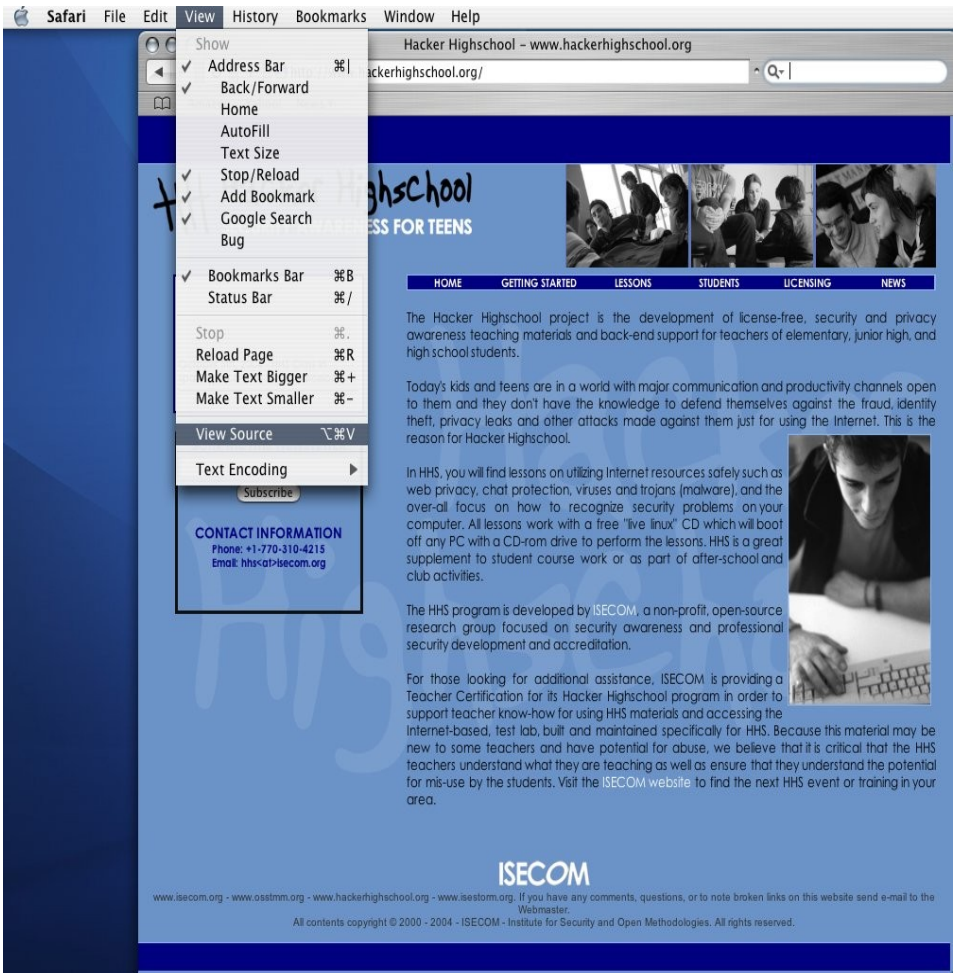

Illustrazione 1Menu Visualizza

Il risultato dovrebbe essere qualcosa molto simile a questo:

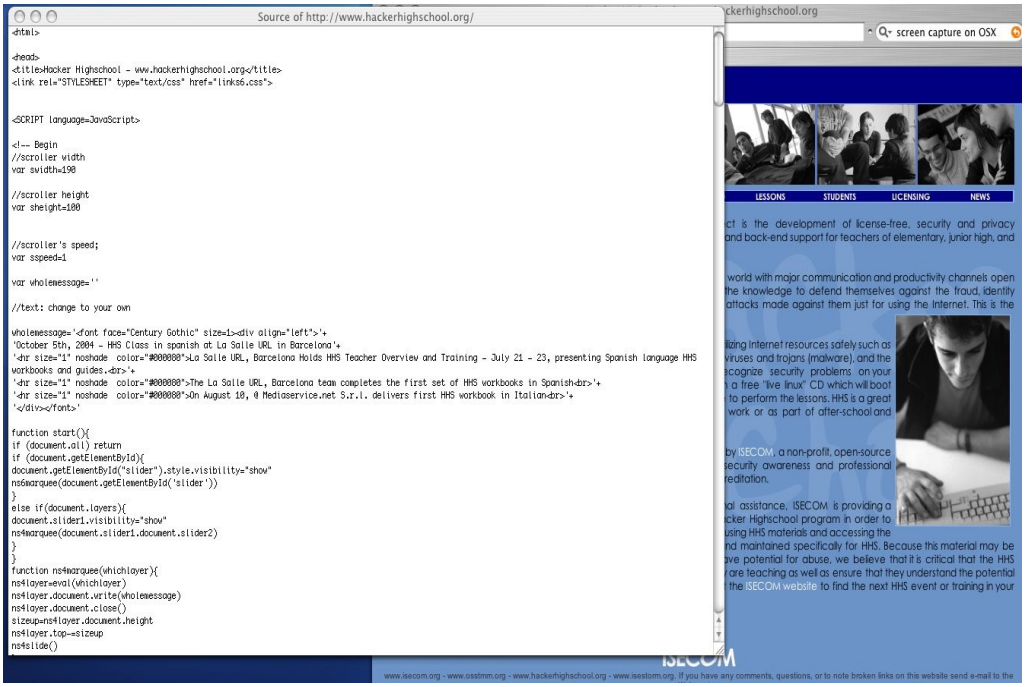

Illustrazione 2Sorgente visualizzato nell'editor di testo

Il codice HTML è visibile a chiunque in un browser web. Questo è il motivo per cui è molto importante non cercare di nascondere password o informazioni importanti nel codice sorgente HTML. Come potete vedere non è molto segreto.

10.3.3 Link

I link (o hyper-link) sono realmente il cuore della costruzione di pagine HTML page. La maggior forza dell'HTML è la sua abilità di collegare altri documenti. Un link, nel contesto dell'HTML, è espresso come www.yahoo.com Questo link apparirà come www.yahoo.com sul vostro sito web. Questo collega i visitatori del vostro sito a Yahoo.

I link possono essere esaminati e seguiti dai programmi chiamati link checker. Questi programmi cercano nel codice sorgente i tag e creano un file o un indice con i collegamenti trovati. Gli spammer usano spesso questa tecnica per cercare indirizzi di posta o moduli di contatto che usano per inviare le loro email. I link checkers possono anche essere usati per cercare all'interno del vostro sito web link "rotti" o link che non portano da nessuna parte. Questo può accadere anche in siti relativamente piccoli.

Esercizio 1: Creare un link

Create un link alla www.hackerhighschool.org che visualizzi Hacker High School sulla vostra pagina web.

Esercizio bonus: Utilizzate lo strumento

1.1.Trovate e scaricate un programma di verifica collegamenti

2.2.Eseguite tale programma sulla www.hackerhighschool.org e documentate quanti link rotti trovate.

10.3.4 Metodi proxy per la manipolazione di applicazioni web

Un proxy HTTP lato server serve come "uomo nel mezzo" tra un server web e un client web (browser). Intercetta e effettua il log di tutte le connessioni tra essi e in alcuni casi manipola quella richiesta di dati per verificare come risponde il server. Questo può essere utile per testare le applicazioni rispetto a vari script di attacco cross-site (fornire il link di riferimento qui), attacchi di SQL Injection e qualunque altro attacco di tipo richiesta diretta. Un'utilità di verifica proxy (SpikeProxy, WebProxy, etc), vi fornirà la maggior parte di questi test. Mentre alcuni di loro effettuano controlli automatici, imparerete rapidamente che è un sostituto debole di una persona dietro tali strumenti.

Esercizio 1: Scegliete il vostro software

1.1. Scaricate uno strumento proxy

2.2. Installate il software seguendo le istruzioni nel file README

3.3. Modificate le impostazioni del vostro browser facendolo puntare al nuovo proxy

Questo è generalmente la porta 8080 su localhost per questi strumenti, ma leggete le istruzioni per esserne sicuri.

Una volta installato il server proxy e aver fatto puntare il browser al proxy, navigate nel sito che state testando. Ricordate di accertarvi di utilizzare un sito web che avete il permesso di testare. dopo aver navigato visualizzate la pagina di amministratore del proxy (per SpikeProxy, è http://www.immunitysec.com/resources-freesoftware.shtml) e iniziate a testare il sito. Dall'interfaccia di amministratore trovate lo strumento metodi di attacco di forza bruta dell'autenticazione del sito o di ricerca di cross-site scripting. (vi raccomandiamo di utilizzare Mozilla o Firefox e http://livehttpheaders.mozdev.org/ e http://addneditcookies.mozdev.org/ insieme per modificare le intestazioni e i cooky al volo senza dover utilizzare una porta separata per il proxy. Non solo semplifica le cose, ma è anche un insieme di strumenti più potenti come insegnamo nell'ISECOM OSSTMM Professional Security Tester class (OPST). Ma dal momento che dovremo imparare a predisporre proxy per altre cose, come filtri ad e spam, privacy filters, etc. pensiamo che dovreste impostarne uno realmente e Spike è un buon strumento da provare.)

Un server proxy può essere un buon strumento per aiutarvi a determinare quanto sia solida un'applicazione web. Per test di penetrazione o ricerca vulnerabilità, dovete avere un buon

strumento proxy. Sono disponibili molti tutorial dettagliati sull'uso di SpikeProxy nel sito http://www.immunitysec.com/resources-papers.shtml.

10.4 Proteggere il vostro server

Ci sono varie azioni che possono essere prese per proteggere il vostro server. Queste includono assicurarsi che il vostro software sia sempre aggiornato e che siano state applicate le patch con gli aggiornamenti relativi alla sicurezza disponibili dal produttore. Questo include che il vostro sistema operativo e server web siano aggiornati. In più Firewall e Intrusion detections systems possono aiutare a proteggere il vostro server, come discusso nel seguito.

10.4.1 Firewall

I firewall in origine erano muri tagliafuoco utilizzati come barriere per prevenire il propagarsi del fuoco, ad esempio tra unità di appartamenti all'interno di un edificio. Lo stesso termine è utilizzato per sistemi (hardware e software) che cercano di prevenire l'accesso non autorizzato alle informazioni di un'organizzazione. I firewall sono guardie di sicurezza che, in base a certe regole, consentono o negano l'accesso del traffico che entra o lascia il sistema di un'organizzazione (casa). Sono guardie di sicurezza importanti che cercano di prevenire attacchi ad un'organizzazione da utenti interni o esterni. E' il primo ed il più importante cancello tra sistemi interni ed esterni.

I firewall sono generalmente posti tra Internet e il sistema informativo di un'organizzazione. L'amministratore del firewall configura il firewall con regole che permettono o negano l'ingresso o l'uscita dei pacchetti dall'organizzazione.

Le regole sono fatte utilizzando una combinazione di indirizzi IP Internet Protocol e Porte; tali regole sono fatte in base alle necessità dell'organizzazione ad esempio in una scuola agli studenti l'accesso può essere consentito in base alla carta di identità.

Le regole di sicurezza in una scuola potrebbero consentire l'accesso alle persone che posseggono una carta di identità valida e negarlo a tutti gli altri. Tuttavia le guardie di sicurezza potrebbero avere un altra regola: consentire a chiunque di uscire tranne i bambini piccoli a meno che non siano accompagnati da un adulto. Un sistema simile è adatto ad un firewall a seconda della natura dell'organizzazione, criticità delle informazioni, costo della sicurezza, politica di sicurezza e verifichi rischi.

Il firewall esattamente come una guardia di sicurezza non può entrare nel merito del contenuto informativo di un pacchetto: esattamente come una guardia consente l'accesso a tutte le persone con un documento di identità valido indipendentemente dalla natura della persona. I firewall consentono l'ingresso o l'uscita principalmente in base all'indirizzo IP e al numero di porta. Quindi un ingresso o uscita è possibile mascherando l'indirizzo IP o la porta. Per mitigare il rischio, le organizzazioni utilizzano Intrusion Detection System, che sono spiegati nella sezione successiva.

Esistono vari tipi di firewall a seconda delle sue caratteristiche: packet filter (opera sui pacchetti IP), stateful firewall (opera in base allo stato di una connessione) o application firewall (che utilizzano il proxy).

Un esempio di regola di un firewall potrebbe essere: blocca l'accesso proveniente dall'indirizzo TCP address 200.224.54.253 sulla porta 135 (esempio immaginario); tale regola

indica ad un computer connesso ad Internet di bloccare qualunque traffico proveniente da computer con indirizzo IP 200.224.54.253 che utilizza la porta 135.

Attività importanti collegate ai firewall sono la configurazione iniziale (creare le regole iniziali), il mantenimento del sistema (aggiunta o modifica nell'ambiente), esame dei log di audit, azioni a seguito di allarmi e testing della configurazione.

10.4.2 Intrusion Detection System (IDS)

Immaginate una scuola che abbia specifiche guardie di sicurezza; come possono le autorità rilevare l'ingresso di persone non autorizzate? Le autorità installerebbero sistema d'allarme che suonerebbero all'ingresso di persone non autorizzate. Questo è esattamente la funzione di un intrusion detection system nell'ambito dei computer. Firewall (guardie di sicurezza o recinti) e IDS (sistemi d'allarme o guardie di perlustrazione) lavorano insieme, mentre i firewall regolano ingressi e uscite, gli IDS consentono o negano gli accessi non autorizzati.

In che modo sono utili gli IDS? Esattamente come i sistemi di allarme, gli IDS avvertono le persone autorizzate (campanelli di allarme) che un pacchetto autorizzato è entrato o uscito. Inoltre un IDS può istantaneamente fermare tale accesso o utente dall'ingresso o uscita disabilitando l'utente o l'accesso. Può anche attivare alcuni altri script; gli IDS possono ad esempio prevenire o ridurre l'accesso da un computer o gruppi di computer.

Gli IDS possono essere di tipo host based o network based; gli IDS host based sono utilizzati su computer singoli mentre gli IDS network sono usati tra computer. Gli IDS host based possono essere usati per rilevare, avvisare o regolare attività abnorme su computer critici; network IDS è similmente usato relativamente al traffico tra computer. Gli IDS così possono usati per rilevare attività abnormi.

Gli IDS esattamente come guardie di pattugliamento monitorano regolarmente il traffico di rete per rilevare qualunque anormalità come traffico intenso da alcuni computer o attività non usuale su un server, utenti connessi ad un'applicazione e coinvolti in attività maliziose. Una volta rilevata la deviazione, gli IDS agiscono in base alla regola stabilita dall'amministratore quale invio di un allarme, memorizzazione dell'intrusione in un log di audit, blocco dell'utente da qualunque attività o generazione di uno script per iniziare una sequenza di attività. Gli IDS possono anche rilevare la deviazione in base ai propri database di firme – ogni deviazione rispetto alla firma è rilevata e gestita, questo è simile ad un anti-virus. Gli IDS sono anche usati per rilevare qualunque attività su risorse critiche o per legalità, esaminando quietamente il sospetto.

Esercizi:

1.1. In una organizzazione per la sicurezza di un sistema informativo sono necessari sia i firewall che gli Intrusion Detection System? Se sì perchè? Se no perchè?

2.2. Pensate a un esempio di utilizzo specifico di regole di firewall nella guardiola di ingresso in una scuola: è necessario l'accesso ad Internet? Se no come verrebbe rafforzato tale ruolo?

3.3. Può uno studente accedere al database che contiene le informazioni complete sui voti degli esami di tutti gli studenti? Come si può controllare? Come si può rilevare nel caso dell'accesso da Internet non autorizzato?

10.5 Comunicazioni sicure

Generalmente il concetto associato alle comunicazioni sicure è il processo di sistemi di computer che creano sicurezza e riducono i rischi. Per le comunicazioni elettroniche, sono necessari tre requisiti per assicurare la sicurezza. A) Autenticità b) Integrità c) Non ripudio.

Autenticità: questo concetto ha a che fare con l'assicurazione che la sorgente di una comunicazione sia chi dichiara di essere. Non è difficile falsificare posta elettronica o variare leggermente il nome di una pagina web, e così ridirigere gli utenti, ad esempio http://www.diisney.com sembra essere la pagina web di disney, ma ha due lettere "i" e può confondere. In questo caso venire trasferiti ad un sito di gioco d'azzardo e le comunicazioni non sono sicure.

Integrità: questa proprietà indica che ciò che è stato inviato è esattamente ciò che arriva a destinazione e che non ha subito alterazioni (volontarie o non) durante il trasferimento.

Non ripudio: se le condizioni di autenticità e autenticità sono soddisfatte, il non-ripudio significa che il mittente non può negare di aver inviato la comunicazione elettronica.

Ad esempio, se un sito web mi assegna un premio e lo posso provare - cioè se un sito web mi invia un buono sconto e io verifico che il sito web è autentico e che nessuno ha manipolato l'informazione nel percorso, il sito non può negare che il buono sia stato inviato.

Il modulo utilizzato per assicurare queste condizioni da un sito web è chiamato certificato elettronico.

Mantenere le condizioni di sicurezza ci dà tranquillità nelle comunicazioni elettroniche e consente di assicurare il principio di privacy nel cyberspazio.

10.5.1 Privacy and Confidentiality

Privacy e riservatezza

La maggior parte dei siti web ricevono alcune informazioni da chi si naviga – o esplicitamente tramite la compilazione di moduli opportuni, o tramite metodi nascosti come i cookies o anche registri di navigazione. Queste informazioni possono essere utili e ragionevoli – come ricordare le vostre preferenze di libri su Amazon.com e, quindi, per assicurare la sicurezza alla persona che naviga, molti siti hanno stabilito dichiarazioni di Privacy e Riservatezza.

Privacy si riferisce a tenere vostre le informazioni – o limitarle alla famiglia stretta o ai vostri amici, o ai vostri contatti, ma al più, a coloro cui avete consentito di condividere le informazioni. Nessuno vuole le proprie informazioni condivise dovunque senza controllo. Per questa ragione ci sono materie dichiarate private, cioè di distribuzione ristretta.

D'altro canto, riservato significa che l'informazione resterà segreta, ma questo dal punto di vista della persona che riceve quell'informazione.

Ad esempio, se desiderate un prezzo, ma non volete che la vostra informazione venga distribuita, dichiarate che questa informazione è privata, autorizzate l'informazione a poche

persone, e queste mantengono la riservatezza. Se, per qualche ragione, vi viene richiesto espressamente quel prezzo e voi rispondete che volete che quell'informazione rimanga confidenziale, questo deve essere: chi riceverà l'informazione la dovrà tenere riservata.

Potremmo generalizzare la definizione di riservatezza come "l'informazione ricevuta sotto la condizione di privacy". La manterrò come se fosse una mia informazione privata". E' necessario dichiarare le condizioni di privacy della gestione dell'informazione per dare le assicurazioni elementari di sicurezza.

Inoltre è raccomandato che si leggano le condizioni stabilite dal sito web che visitate nella regolamentazione di sicurezza.

Esercizio:

1. Riesaminate le condizioni di privacy dei fornitori mondiali di WebMail: Google e Hotmail e di industrie quali General Motors http://www.gm.com/privacy/index.html. Sono le stesse? Di queste, chi diffonderà le informazioni che fornisco? Quali misure sarò in grado di prendere se essi non osserveranno queste regole?

10.5.2 Sapere se si sta comunicando in maniera sicura

Anche sotto le condizioni di Privacy e Riservatezza, qualcuno potrebbe ancora intercettare le comunicazioni. Per fornire le condizioni discusse all'inizio di questa sezione, un livello di sicurezza precedentemente esaminato, SSL, utilizza certificati digitali per stabilire connessioni sicure (cioè che soddisfino le condizioni di autenticità, integrità e non ripudio) e provvede un livello di crittografia nelle comunicazioni (questo per nascondere le informazioni in modo tale che chiunque prenda parte alla comunicazione non possa accedervi perchè il messaggio è criptato in modo tale che solo il mittente e il destinatario, con certificati corretti, siano in grado di comprenderlo). Questo livello è chiamato Security Socket Layer, SSL, ed è visibile attraverso due elementi all'interno del web browser.

Una comunicazione è considerata sicura quando l'indirizzo web (URL) cambia da HTTP a https, questo cambiamento modifica anche la porta di comunicazione da 80 a 443. Oltre a ciò, nella barra più in basso del browser, appare un lucchetto chiuso, che indica le condizioni di sicurezza nella comunicazione.

Se posizionate il mouse su questo lucchetto, apparirà un messaggio che specifica il numero di bit usati per fornire la comunicazione (il livello di crittografia), utilizzato, 128 bit sono il livello di crittografia raccomandato. Questo significa che verrà usato un numero che può essere rappresentato in 128 bit per basare la comunicazione.

Esiste un tipo di trucco chiamato phishing (http://www.antiphishing.org/) in cui viene simulata la pagina Web di una banca (copiano la grafica, in modo tale che i clienti inseriscano i propri dati, confidando che quella sia la loro banca, nonostante non lo sia). Per evitare queste situazioni, l'autenticità del sito andrebbe verificata e andrebbe verificato che la comunicazione fosse sicura (https e il lucchetto chiuso), e al meglio della vostra conoscenza verificato il certificato.

10.6 Metodi di Verifica

A questo punto, avete avuto l'opportunità di conoscere le basi della sicurezza sul web, gli aspetti principali legati ad alcune vulnerabilità che si incontrano comunemente nei server web usati per ospitare i vari siti con cui interagiamo normalmente in Internet tramite browser, e i vari difetti nello sviluppo di applicazioni web e/o la privacy degli utenti in generale.

D'altro lato, avete imparato alcune tecnologie sui cui ci basiamo per proteggere i nostri server e anche la nostra privacy. Tuttavia, probabilmente a questo punto, vi starete ponendo questioni quali: sono sicuro ora di aver preso le misure adeguate? Il mio sistema è sicuro? Gli sviluppatori che hanno programmato alcune funzionalità che ho usato nel mio sito web, si sono presi cura di assicurare gli aspetti della sicurezza? Come posso verificare tali aspetti?

Come probabilmente avrete pensato, non è sufficiente applicare gli aggiornamenti del produttore o aver fiducia nelle buone intenzioni dello sviluppatore, quando è in gioco la vostra sicurezza o privacy. Nel passato, si sono verificati vari casi in cui le patch di un produttore correggevano una vulnerabilità, ma causavano un altro problema nel sistema, o una volta applicata la patch veniva scoperta un'altra vulnerabilità. Per questo e altre ragioni, dovrete considerare che è assolutamente necessario verificare frequentemente i sistemi implementati per mantenere il sistema sicuro.

Fortunatamente, molte persone hanno sviluppato nel loro tempo libero, alcuni "Metodi di Verifica", la maggior parte dei quali sono disponibili liberamente, in modo tale che tutti possano trarre vantaggio dai benefici del loro utilizzo. Si basano sull'esperienza di centinaia di professionisti e includono varie "buone pratiche" riguardanti la tecnologia sicura. Quindi raccomandiamo che adottiate queste metodologie al momento di effettuare il vostro compito di verifica.

Un esempio di queste, OSSTMM, verrà discussa brevemente nel seguito.

10.6.1 OSSTMM

OSSTMM, acronimo di "Open Source Security Testing Manual Methodology" è una metodologia di verifica sicurezza molto diffusa. Come descritto nell'introduzione, nonostante vengano menzionati certi test individuali, questi non sono particolarmente rivoluzionari, una metodologia complessiva rappresenta uno standard di riferimento essenziale, per chiunque voglia effettuare test di sicurezza in modo ordinato e con qualità professionale. OSSTMM, è suddiviso in varie sezioni. E' possibile identificare al suo interno una serie di moduli di verifica specifici, attraverso cui viene testato ogni aspetto di sicurezza e integrato con le operazioni necessarie per assicurare sicurezza.

Queste sezioni includono: Sicurezza Personale, Sicurezza delle Reti Dati, Sicurezza delle Telecomunicazioni, Sicurezza delle Comunicazioni Wireless, e Sicurezza Personale e la sezione di dettagli di sicurezza di questa metodologia dal punto di vista di QUALI test vanno eseguiti, PERCHE' e QUANDO.

OSSTMM dettaglia gli ambiti tecnici e le operazioni tradizionali di sicurezza, ma, e questo è forse uno degli aspetti più significativi, non il test esatto, piuttosto presenta, cosa dovrebbe

essere testato, la forma in cui i risultati del test dovrebbero essere presentati/visualizzati, le regole da seguire per chi effettua i test per assicurare i risultati migliori, e anche incorpora il concetto di metrica di sicurezza con RAVs (Risk Assessment Values) per identificare con un numero reale quanta sicurezza avete. OSSTMM è un documento per professionisti, ma non è mai troppo presto per cercare di capire e imparare come lavora. I concetti sono molto completi ed è scritto in una maniera molto facile da comprendere.

Esercizi

1.Applicare delle patch è oggi un problema comune: gli amministratori web devono continuamente applicare patch al codice ogni volta che vengono scoperte nuove vulnerabilità. Cercate un caso in cui si sia verificato un problema installando una nuova patch. Discutete le possibilità e conseguenze che un amministratore deve affrontare quando, dovendo installare una nuova patch, realizza che questo aprirà una vulnerabilità nel suo sistema che è già stata risolta. Dovrebbe essere installata ancora la patch? In relazione a questo problema, è importante o no avere a disposizione il codice sorgente?

2.Collegatevi al sito http://cve.mitre.org e cercate CVE. Inserite il nome di un server web (ad esempio Apache) nel campo di ricerca. Quando è stata scoperta l'ultima vulnerabilità? Quanto frequentemente sono state scoperte nuove vulnerabilità (settimanalmente, mensilmente, ...)? In riferimento alla prima domanda applicare una patch è una soluzione realistica per la sicurezza? Perchè sì o perchè no? Quali altre misure di sicurezza possono essere adottate se non si vuole giocare al gatto e al topo con le patch?

3.Effettuate il download di una copia di OSSTMM e rivedete i concetti della metodologia. Quali aspetti enfatizzereste da questa metodologia? Come pensate che questa metodologia possa essere integrata con le vostre verifiche di sicurezza?

4.Cosa potete ricavare dalla RAV?

Letture di approfondimento

http://www.osstmm.org

http://www.oreilly.com/catalog/websec2/chapter/ch08.html

http://www.w3.org/Security/Faq/

http://www.privacyalliance.org/

http://www.perl.com/pub/a/2002/02/20/css.html

http://www.oreilly.com/catalog/webprivp3p/chapter/ch01.pdf

http://www.defenselink.mil/specials/websecurity/

http://www.epic.org/

http://www.cgisecurity.com/

http://www.eff.org/privnow/

Nel seguito sono elencati alcuni siti dove è possibile trovare informazioni sulla creazione di pagine web o HTML in generale.

http://www.htmlgoodies.com/
http://www.htmlhelp.com/
http://www.w3schools.com/