Cognome:_____ Nome: _____ Matricola:_____

Sicurezza dei sistemi informatici e delle reti – 23 giugno 2017 – 6 CFU

òTempo a disposizione: 70 minuti. Libri e appunti chiusi. Vietato comunicare con chiunque. Vietato l'uso di smartphone/watch, calcolatrici e affini.

1. **PKI e crittografia.**

1.1. Descrivi il concetto di certificato in una PKI e i principali campi dei certificati X509.

vedi materiale didattico

1.2. Elenca brevemente almeno tre aspetti deboli dell'uso delle PKI nel web.

1. implementazioni dei browser poco attente alle revoche dei certificati
2. incompetenza dell'utente
3. si ripone fiducia nei root certificates configurati nel browser

(questi sono solo 3 esempi, vedi materiale didattico)

1.3. Descrivi il concetto di firma elettronica e spiega l'attacco alla firma elettronica che sfrutta il paradosso del compleanno.

vedi materiale didattico

2. **Principi di progettazione.** Discuti brevemente la sinergia o l'antagonismo tra le seguenti coppie di principi di progettazione visti a lezione.

2.1. Eterogeneità vs. semplicità di progetto

Sono due principi antagonisti: un sistema eterogeneo (che sfrutta prodotti di più vendor) risente meno della presenza di vulnerabilità su prodotti di un solo vendo rispetto ad un sistema non eterogeneo. Tuttavia l'eterogeneità è complica il progetto e anche la gestione del sistema.

2.2. Usabilità vs. default sicuri

Sono due principi antagonisti: un prodotto con un default sicuro potrebbe richiedere la configurazione esplicita di permessi per ammettere certe operazioni (es. un firewall che inizialmente non lascia passare alcuna connessione). La necessità di effettuare queste configurazioni ne diminuisce l'usabilità.

2.3. Isolamento vs. mediazioni completa

Sono due principi sinergici: se tutte le interazioni tra soggetti e oggetti sono mediate da un security kernel, è facile configurare il security kernel per creare ambiti isolati in cui gli oggetti di un certo ambito possono essere accedute solo da soggetti di tale ambito.

Cognome:_____ Nome: _____ Matricola:_____

Sicurezza dei sistemi informatici e delle reti – 23 giugno 2017 – 6 CFU

3. Autenticazione con password.

3.1. Che significa avere una password "easy-to-guess" (o in altri termini *debole*)? Fornisci degli esempi di metodi per creare password che pur apparendo non banali risultano comunque "easy-to-guess".

Sono password derivate mediante semplici regole da parole di dizionari standard e cifre. Esempi: una parole seguita/preceduta da cirfre (bella2005), una concatenzazione di due parole (bellaprova), consonanti maisucole vocali minuscole (BeLLaPRoVa,), sostituzione di lettere con cifre (83LLaPR0Va)

3.2. Supponi che un account accessibile da Internet abbia una password debole, descrivi l'attacco on-line e le contromisure a livello di configurazione di sistema per ridurre la probabilità di riuscita dell'attacco anche in caso di password debole.

L'attacco on-line consiste nel provare consecutivamente le password deboli. Le contromisure tipiche sono: blocco dell'account dopo un certo numero di tentativi falliti, tempo di risposta crescente dopo ciascun errore. Un altro approccio tipico è fare in modo che non ci siano password troppo deboli simulando periodicamente degli attacchi off-line da parte dell'amministratore di sistema.

3.3. Considera ora l'attacco off-line ad un database di utenti in cui è memorizzato l'hash crittografico di ciascuna password (es. SHA256) senza altra accortezza. Che tecnica può usare un hacker per avere una alta probabilità di invertire l'hash per password anche non-deboli? Descrivila brevemente.

rainbow tables
descrizione: vedi materiale didattico

4. Vulnerabilità di TCP. Descrivi brevemente i seguenti attacchi a TCP spiegando le rispettive vulnerabilità

4.1. Attacco alla confidenzialità della sessione.

Vulnerabilità
trasmissione in chiaro

Attacco
sniffing

4.2. DoS tramite TCP Reset

Vulnerabilità
autenticazione della sorgente mancante

Attacco
invio di tcp reset a sessioni remote

Session hijacking (cioè furto di sessione già aperta).

Vulnerabilità

autenticazione della sorgente mancante

Attacco
l'attaccante esclude una delle due parti e ne prende il suo posto nella comunicazione

4.3. Man-in-the-middle attivo (cioè modifica dei dati nel flusso).

Vulnerabilità
autenticazione della sorgente e verifica di integrità dei dati mancanti

Attacco
Modifica dei dati in transito, potrebbe richiedere anche cambiare i numeri di sequenza

5. **Sicurezza in ambiente Windows.** Considera la funzionalità di controllo di accesso nei sistemi Windows.

5.1. Qual è il suo input? (cita le strutture dati coinvolte)

access token – contenente le credenziali del processo
security descriptor – diritti di accesso (ACL) associati a ciascun oggetto
access mask – elenca le operazioni che si richiede di poter svolgere sull'oggetto

5.2. Qual è il suo output?

"accesso consentito" o "accesso negato"

5.3. Descrivi l'algoritmo di discretionary access control disponibile nei sistemi Windows.

vedi materiale didattico

6. **Sicurezza del codice.** Considera un server S su cui sono installati una web application scritta in PHP. Rispondi alle seguenti domande.

6.1. La web application è accessibile da Internet via HTTP, gli accessi arrivano attraverso un firewall che non fa deep packet inspection, l'interprete PHP del web server è configurato per non applicare nessuna elaborazione sui parametri forniti dall'utente. La form di login contiene il seguente codice html/php:

```
<title> Autenticazione per <?php echo $_GET["t"] ?> </title>
```

dove `t` è un parametro che è passato nell'URL come nel seguente esempio

```
http://esempio.it/index.php?t=Servizi%20di%20base
```

Noti una o più vulnerabilità? Spiega.
Si. Si tratta di un Cross Site Sripting (XSS). Un attaccate può inserire uno script come valore del parametro t e questo sarà riportato nella pagina html scaricata dal browser ed eseguito.

Se vulnerabile, pensi che ciò rappresenti una problema di sicurezza? Discuti brevemente.
E' un problema di sicurezza se l'url può essere scelto a piacimento da un soggetto che può avere interesse a modificare il comportamento del sito. Nel caso in esame l'input viene da Internet, quindi sì.

6.2. Una volta loggati gli utenti possono eseguire una ricerca in un database. Il db è realizzato con mysql e accessibile in php tramite l'oggetto mysqli il cui utilizzo dovrebbe essere chiaro dal codice stesso. Ricorda che in php l'operatore che concatena le stringhe è il punto. Il codice che processa la ricerca è il seguente

```
<h3> Risultati </h3>
<table> <?php
$query = "SELECT descr FROM art WHERE descr LIKE '%". $_GET["q"] ."%' ";
$result = $mysqli->multi_query($query);
while ($row = $result->fetch_array()) {
   echo "<tr><td>". $row["descr"] ."</td></tr>";
}
?> </table>
```

Noti una o più vulnerabilità? Spiega.
Due vulnerabilità.

SQL injection: il parametro q viene usato per comporre una query senza subire prima alcun processamento. E' possibile cambiarne la semantica con opportuna volorizzazione, esempi
' or 1==1
'; show tables; -- "

Sospetto XSS persistente: il contenuto del campo descr viene inserito nella pagina web senza alcun processamento. Tale campo potrebbe contenere degli script. L'effettiva importanza di questa vulnerabilità dipende dal codice che valorizza il campo descr. Se tale valorizzazione prende input da utente e non prende alcuna precauzione siamo in presenza di un XSS persistente.

Se vulnerabile, pensi che ciò rappresenti una problema di sicurezza? Discuti brevemente.
Come sopra.