Cognome:_____ Nome: _____ Matricola:_____

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

Usa questa pagina per la brutta, staccala, non consegnarla.

Cognome:_____ Nome: _____ Matricola:_____

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

Usa questa pagina per la brutta, staccala, non consegnarla.

Cognome:_____ Nome: _____ Matricola:_____

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

Tempo 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.

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

1.

2.

3.

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

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

2.2. Usabilità vs. default sicuri

2.3. Isolamento vs. mediazioni completa

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".

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.

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.

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

4.1. Attacco alla confidenzialità della sessione.

Vulnerabilità

Attacco

4.2. DoS tramite TCP Reset

Vulnerabilità Attacco

Session hijacking (cioè furto di sessione già aperta).

Vulnerabilità Attacco

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

Vulnerabilità Attacco

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)

5.2. Qual è il suo output?

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

Cognome:_____ Nome: _____ Matricola:_____

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

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.

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

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.

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