Cognome:_____ Nome: _____ Matricola:_____

Sicurezza dei sistemi informatici e delle reti – 18 luglio 2014

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

1. **Sicurezza del codice.** Analizza la sicurezza nei seguenti stralci di codice C relativi alla esecuzione di un comando di copia i cui parametri sono passati come argomenti dell'eseguibile.

1.1.

```
int main(int argc, char** argv) {
      char buffer[2000];
      int j;
      strcpy(buffer, "/bin/cp");
      for( j=1; j<argc; j++) {
           strcat(buffer, " ");
           strcat(buffer, argv[i]);
      }
      system(buffer); /*esegue il contenuto di buffer nella shell */
   }
```

buffer overflow: la concatenazione degli argomenti può essere maggiore di 2000 caratteri
system: non vi è alcun controllo sul comando eseguito tramite system e creato dall'input

1.2.

```
int main(int argc, char** argv) {
      char buffer[2000];
      if (strlen(argv[1])>1999)
           {/*errore: argomento troppo lungo*/}
      strcpy(buffer, "/bin/cp");
      strcat(buffer, argv[1]);
      system(buffer); /*esegue il contenuto di buffer nella shell */
   }
```

buffer overflow: il controllo sulla lunghezza di argv[1] non è fatto bene si può ancora avere overflow
system: vedi sopra

1.3.

```
int main(int argc, char** argv) {
     /* esegue cp con gli stessi argomenti e lo stesso ambiente del padre */
      execve("/bin/cp", argv, environ);
                          /*environ punta all'ambiente corrente*/
   }
```

execve non ha i problemi di system, tuttavia se cp avesse dei bug non vi alcuna funzionalità di wrapping per evitare che cp possa essere sfruttato a fini malevoli, tramite l'ambiente o gli argomenti.

Cognome:_____ **Nome:** _____ **Matricola:**_____

Sicurezza dei sistemi informatici e delle reti – 18 luglio 2014

1.4.
```
int main(int argc, char** argv) {
      int j;
      if (argc>3)
            {/*errore: troppi argomenti*/}
      for( j=1; j<argc; j++)
         if(argv[j][0]=='-')
            {/*errore: opzione per cp!*/}
      execve("/bin/cp", argv, NULL); /*stessi argomenti ma ambiente nullo*/
   }
```

l'ambiente non viene passato e gli argomenti vengono verificati (almeno un po') per cui è più difficile sfruttare bug di cp a fini malevoli. Manca una verifica sulla lunghezza e sulla ammissibilità dei caratteri nei due argomenti che si possono passare.

2. Sicurezza delle reti.

2.1. Descrivi il DDOS noto come syn-flood.

Che caratteristiche ha il traffico che arriva all'obiettivo dell'attacco?

vedi materiale didattico

Quali sono le risorse saturate?
vedi materiale didattico

2.2. Che ruolo può avere un firewall nella protezione dai syn-flood?

può evitare che sessioni iniziate ma non proseguite vadano a saturare le risorse del server. si presuppone che il firewall sia progettato per essere abbastanza scalabile o usi tecniche tipo syn cookies
vedi materiale didattico

2.3. Descrivi una tecnica per effettuare load balancing su più firewall in modo che traffico relativo alla stessa connessione passi per lo stesso firewall.

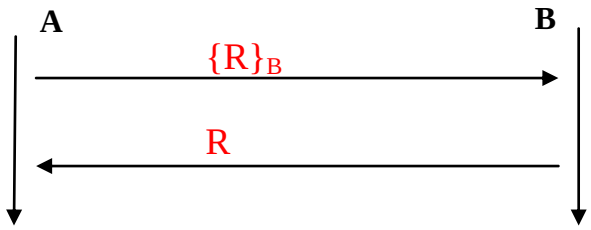
vedi materiale didattico (load balancing con la tecnica degli hash)

Cognome:_____ **Nome:** _____ **Matricola:**_____

Sicurezza dei sistemi informatici e delle reti – 18 luglio 2014

3. Protocolli crittografici.

3.1. Supponi che un server B sia dotato di una chiave privata. Un client A, in possesso della relativa chiave pubblica, deve autenticare B. Mostra **il più semplice** protocollo di autenticazione basato sull'approccio challenge-response **in cui il server decifra il challenge**.



R: challenge (nonce) generato da A

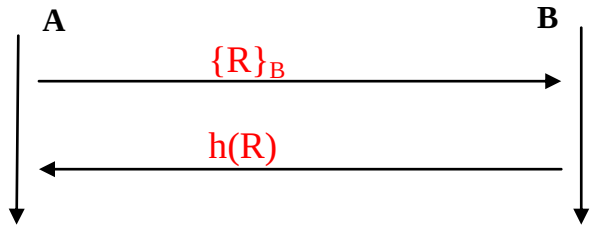
3.2. Considera il protocollo al punto 1. Che tipo di attacco crittoanalitico (tra ciphertext only, known plaintext, chosen plaintext) si puo instaurare facendo solo richieste lecite a B?

know plaintext

3.3. Considera il protocollo al punto 1. Se un attaccante possiede un messaggio cifrato con la chiave pubblica di B e ne vuole conoscere il contenuto come può sfruttare B?

basta inviarlo a B, B risponderà con la versione decifrata

3.4. Fornisci una variante del protocollo al punto 1 che non abbia i problemi elencati.



R: challenge (nonce) generato da A
h(.): una funzione di hash crittografico.

4. Principi di progettazione

4.1. Progetto aperto. Perché è considerato un principio importante?

vedi materiale didattico

4.2. Default sicuri. Perché è considerato un principio importante?

vedi materiale didattico

4.3. Mediazione completa. Perché è considerato un principio importante?

vedi materiale didattico

4.4. Nell'ambito dei sistemi operativi come si realizza il principio di mediazione completa?

tutte le chiamate di sistema devono avere un controllo di accesso.
Meglio se il codice di controllo è fattorizzato in un reference monitor. Particolare attenzione va posto a tecniche di ottimizzazione che prevedono il check solo sulla open e non su ciascuna read/write. In questo caso è possibile ancora adottare tali tecniche ma bisogna fare attenzione alla semantica delle modifiche ai permessi dei file e a ciò che accade ai file già aperti

5. Pianificazione

5.1. Perché è importante avere un piano di sicurezza?

vedi materiale didattico

5.2. Quali sono gli obiettivi dell' "analisi del rischio"

valutazione di impatto e probabilità di evento avverso (vedi materiale didattico)

5.3. In che rapporto è, in un piano di sicurezza, la parte relativa all'analisi del rischio con "l'analisi dello stato attuale" e con le "contromisure"?

Con l'analisi dello stato attuale:

l'analisi dello stato attuale è in sostanza un inventario, l'analisi dei rischi deve essere fatta considerando le voci di tali inventario.

Con le contromisure:

le contromisure vanno a mitigare e fronteggiare rischi (almeno quelli più importanti). è bene valutare anche il rischio residuo a valle dell'adozione della contromisura.

6. Sicurezza in ambiente Windows. Supponi di voler idealmente creare una matrice di accesso che rappresenti lo stato di sicurezza di un sistema Windows.

6.1. Cosa identificheresti come soggetti?

i processi o gli access token, o i sid

6.2. Cosa identificheresti come oggetti?

gli executive objects

6.3. Che cosa identificheresti come diritti?

per ogni casella della matrice, l'access mask più generica (quella che chiede più diritti) che può essere accettata in base alle ACL dell'executive object e dell'access token (colonna e riga). Vedi algoritmo di controllo di accesso.

Cognome:_____ Nome: _____ Matricola:_____

Sicurezza dei sistemi informatici e delle reti – 18 luglio 2014