Minacce alla sicurezza dei programmi

Lezione 9 di Sicurezza dei sistemi informatici 1

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2007-8

Indice

1. Minacce alla sicurezza dei programmi
2. buffer overflow intenzionale
3. mediazione incompleta
4. difetto per serializzazione (o diacronia)
5. combinazioni di difetti dei programmi
6. codice maligno: tassonomia
7. virus: diffusione, controllo, ospiti
8. virus: schemi di definizione
9. prevenzione delle infezioni virali

buffer overflow intenzionale

come già accennato nella lezione precedente, gli effetti del buffer overflow dipendono dall'area in cui si trova la locazione di memoria a cui si tenta l'accesso erroneo

in particolare, se un sottoprogramma in esecuzione riesce a sovrascrivere l'indirizzo di ritorno nello stack di sistema, potrà in tal modo trasferire il controllo a del codice che sia stato caricato (tipicamente dallo stesso sottoprogramma) a tale indirizzo

> si parla in tal caso di **stack overflow**, così intendendo un trabocco **sullo** stack

il buffer overflow è dunque anche una tecnica di attacco per l'introduzione ed esecuzione di **codice maligno** nei sistemi informatici

> le tecniche di attacco che sfruttano vulnerabilità di buffer overflow a tali scopi non sono limitate all'overflow sullo stack, ma possono riguardare lo heap, porte di servizi, parametri in URL, etc.

su questo argomento, tre contributi di studenti di una precedente edizione del corso sono consultabili nella sezione Appunti e dispense del sito dell'insegnamento

mediazione incompleta

malfunzionamenti di programmi possono essere causati da valori erronei dell'input di utente

esistono essenzialmente due modi per impedirli:

> controllo **a posteriori:** valori illeciti dell'input possono venire scartati, con eventuale gestione degli errori da procedure apposite
> **prevenzione:** per il tramite dell'interfaccia di utente del programma

nel secondo caso, il controllo sul lato client del programma consiste nel permettere all'utente solo la **selezione dell'input** all'interno del dominio dei valori leciti

> ad es., ciò può realizzarsi mediante una interfaccia grafica che presenti la selezione da una tabella predefinita, o menu a tendina, etc.

in tal modo, l'interfaccia di utente effettua una **mediazione** dell'input

> si ha **mediazione incompleta** quando la mediazione non impedisce la successiva **modifica** dell'input da parte dell'utente, **prima** della sua trasmissione da parte del client

esempio: si consideri un programma che accetti ordini di acquisto via Web

> l'utente potrà selezionare prodotti e relative quantità, che assieme ai rispettivi prezzi determinano l'importo totale dell'ordine
> si supponga però che i dati dell'ordine, vale a dire codici dei prodotti, quantità e prezzi, vengano codificati dall'interfaccia Web in una URL associata all'ordine, **esposta** nel browser...
> ... l'utente potrà quindi modificare i prezzi dei prodotti e l'importo totale a suo gradimento!

difetto per serializzazione (o diacronia)

una condizione necessaria per la sussistenza di vulnerabilità di mediazione incompleta è il fatto che il controllo di validità dell'input di utente e la sua effettiva trasmissione avvengano in momenti diversi, e che nel lasso di tempo fra questi l'utente abbia la possibilità di modificare l'input

una condizione analoga caratterizza i difetti di **serializzazione**, generalmente dovuti alla **diacronia** fra controllo e uso

più precisamente, i difetti di serializzazione generalmente riguardano il **controllo di accesso** a una risorsa e l'**effettivo accesso** del soggetto della richiesta di accesso

come si vedrà in una lezione successiva, il controllo di accesso consiste nella verifica che il **soggetto** richiedente goda dell'**autorizzazione** necessaria all'accesso a un dato **oggetto** per una data **operazione**

> un difetto di serializzazione può vanificare l'efficacia del controllo di accesso se il soggetto può intervenire sui parametri della richiesta (oggetto e/o operazione) **dopo** l'avvio del controllo di accesso e **prima** dell'accesso effettivo

combinazioni di difetti dei programmi

come si è visto, i tre tipi di difetti considerati, seppur non intenzionali, possono essere abilmente sfruttati per dar luogo a serie minacce alla sicurezza:

> introduzione ed esecuzione di codice maligno
>
> malfunzionamenti di programmi causati da valori illeciti dell'input
>
> violazione del controllo di accesso

minacce ancor più serie sono da attendersi dall'abilità di un attaccante di sfruttare **combinazioni** di tali difetti

ad esempio, un aggressore consapevole della presenza di tali difetti in un sistema informatico può orchestrare un **attacco in più stadi**, in cui potrà:

> forzare l'esecuzione di codice maligno per buffer overflow, che gli permetta di
>
> creare un nuovo utente, ad es. sfruttando una mediazione incompleta, e quindi
>
> assegnare indebiti privilegi al nuovo utente, ad es. grazie a un difetto di serializzazione

codice maligno: tassonomia

virus:
si combina a un programma e propaga copie di sé in altri programmi

cavallo di Troia:
contiene funzionalità nascoste

bomba logica:
innesca un'azione al verificarsi di una condizione logica

bomba a tempo:
innesca un'azione a un dato istante

trapdoor o **backdoor (porta nascosta):**
offre accesso non autorizzato alle funzionalità

worm (verme):
propaga copie di sé attraverso una rete

rabbit (coniglio):
si replica fino all'esaurimento delle risorse

virus: diffusione, controllo, ospiti

una caratteristica del virus è la necessità di un programma **ospite** per assicurarne l'esecuzione
un virus è detto **transitorio** se opera soltanto durante l'esecuzione del programma ospite, altrimenti è detto **residente**

la **diffusione** di un virus dipende dal tipo di programma ospite, e può sfruttare una varietà di mezzi:

virus del **settore di avvio:**
associato al codice di avvio del sistema, si diffonde infettando il settore di avvio del disco

virus **allegato a un messaggio di e-mail:**
particolarmente pericoloso quando il client di e-mail è configurato per l'apertura automatica degli allegati: pratica assolutamente sconsigliata!

virus dei **documenti:**
sebbene i documenti generalmente non siano programmi eseguibili, quelli formattati con editor sofisticati possono contenere macro complesse interpretabili dall'editor, comandi di sistema, etc.: un virus può sfruttare queste possibilità per diffondersi e/o installare codice maligno

un virus può anche diffondersi da un **sito Web**, ad es. inserito in uno **script** interpretato dal browser

un programma infetto da un virus può assumere il controllo in luogo dell'originale sovrascrivendo il codice originale o un puntatore ad esso (nella tabella dei processi, nel file system, etc.)

virus: schemi di definizione

un obiettivo degli autori di virus, essenziale ad assicurarne l'efficacia, è l'**occultamento** della presenza del virus nel programma ospite

tuttavia, nessun virus è invisibile: oltre all'eventuale diversità di **comportamento**

- se il codice virale si aggiunge a quello dell'ospite, la **dimensione** del programma infettato differisce da quella dell'originale
- altrimenti l'infezione deve **sopprimere** una parte del codice dell'ospite, per mantenerne inalterata la dimensione, ma ciò ne provoca comunque diversità di **struttura**

la presenza di un virus può talvolta essere rilevata da tratti tipici della sua esecuzione:

- schema di **esecuzione:** operazioni compiute dal virus per molteplici obiettivi, ad es.: diffusione dell'infezione, occultamento della propria presenza, danni alla sicurezza del sistema
- schema di **trasmissione:** trasduzione del virus attraverso supporti e vie di comunicazione

la rilevazione più affidabile ed efficace della presenza di virus è tuttavia quella effettuabile **prima** che il virus entri in azione, in quanto basata sul suo schema di **memorizzazione:** tratti caratteristici del codice del virus nel codice dell'ospite, che ne costituiscono la **"firma"** o **(schema di) definizione**

si dicono **polimorfi** i virus in grado di assumere molteplici schemi di definizione, equivalenti nel comportamento, al fine di occultare la propria presenza ai programmi di rilevamento (**"antivirus"**)

- una categoria più sofisticata di virus polimorfi impiega la **crittografia** a tal fine, dove il codice virale consta di tre parti: chiave di decifrazione, codice virale crittografato, codice di decifrazione in chiaro

prevenzione delle infezioni virali

non esiste una ricetta di efficacia universale per la prevenzione delli infezioni virali

...

... tuttavia alcune **"buone abitudini"** possono ridurre notevolmente il rischio di incapparvi:

- uso di software di **provenienza affidabile**
- **collaudo** di nuovo software su un **sistema di prova**, prima dell'installazione sul sistema di esercizio
- apertura di **allegati** solo se di **provenienza sicura**
- configurazione appropriata delle impostazioni di **sicurezza del browser**
- **backup** periodico del sistema, dei dati e degli eseguibili potenzialmente attaccabili da virus, **su supporti isolati**
- uso regolare e aggiornamento frequente dei **rilevatori di virus**