Reversing 2
(De)ofuscación

CC5325 - Taller de Hacking Competitivo

Basada en la clase que hizo Diego el año pasado

¿Qué veremos hoy?

- Ofuscación y técnicas
- Deofuscación y algunas técnicas
- Un par de problemas

Ofuscación

- **En computación, entendemos la ofuscación como el acto de cambiar el código (antes o después de la compilación) de un programa, sin cambiar su funcionalidad, pero sí su legibilidad.**
- **Como aclaración, compilar un programa no corresponde a ofuscación, a pesar de que lo que obtenemos se puede considerar menos legible.**
- **Para efectos del ramo no veremos ofuscación de código compilado, aunque algunos problemas de reversing que encuentren podrían tener**

Técnicas de ofuscación: ofuscación básica o renaming

- Consiste en cambiar nombres de variables y funciones con la finalidad de que el código sea menos legible.
- A menudo no sirve de mucho si el código original no es lo suficientemente complejo

```
function a0($f1,$c2){
  $w3=$f1;
  for($r4=0;$r4<$c2;$r4++){
    $w3=0.5*($w3+($f1/$w3));
  }
  return $w3;
}
$f1=$b5[1];
$c2=$b5[2];
$w3=a0($f1,$c2);
echo("$w3\n");
```

Técnicas de ofuscación: ofuscación básica o renaming

```
function a0($f1,$c2){
  $w3=$f1;
  for($r4=0;$r4<$c2;$r4++){
    $w3=0.5*($w3+($f1/$w3));
  }
  return $w3;
}
$f1=$b5[1];
$c2=$b5[2];
$w3=a0($f1,$c2);
echo("$w3\n");
```



```
function a0($x0, $x1){
  $res=$x0;
  for($i=0; $i<$x1; $i++){
    $res=0.5*($res+($x0/$res));
  }
  return $res;
}
$x0=$data[1];
$x1=$data[2];
$res=a0($x0,$x1);
echo("$res\n");
```

Técnicas de ofuscación: eval

- Se usa principalmente en lenguajes interpretados.
- El código se encripta, comprime y codifica en un string o constante, para después ejecutarse usando una función (como eval)

```
from base64 import b64decode
from zlib import decompress

payload =
'eJw1yzEKwkAQBdCr/ExlQPYC4gFsbNLaaHZwPyQ
za2YEc3ttbB+8vtHyIFd/eN2hn65zBrIppn43RsP
FXm8Gk26DHNXqWW6bjKf+n4nqptg1SymYfNVstCc
YiOSyYGXEDwYZv35UJm0='
decoded = b64decode(payload)
decompressed = decompress(decoded)

exec(decompressed)
```

El ejemplo lo pueden ejecutar en
https://www.python.org/shell/

Técnicas de ofuscación: string splitting

- Consiste en esconder el valor de un string en diversas partes.
- Después, estas partes se manipulan para obtener el valor escondido. Para esto, se puede usar concatenación, casting, substrings, codificaciones más o menos raras, etc.
- El ejemplo lo pueden ver en el apunte (acá no cabe bien)

Técnicas de ofuscación: dead branches

- En esta técnica se agregan ramas de ejecución que nunca son ejecutadas (if/else/while/etc).
- Muchas veces también se agrega código que, si bien se ejecuta, no incide en el resultado final de la ejecución, pero sí como ruido al leerla o debuggearla.
- A veces, dependiendo del compilador y su configuración, este puede “borrar” parte del código muerto, pero esto no siempre pasa (y no siempre es posible).

Deofuscación

Evaluación manual

- Se revisa el código manualmente, renombrando variables, funciones y clases, decodificando strings y otras constantes, quitando ramas condicionales que no se ejecutan, etc.
- Se suele ejecutar línea por línea, estilo debugger, para revisar el valor de los diferentes objetos y variables en tiempo de ejecución.
- Se puede aplicar a códigos pequeños y de poca complejidad, ya que usualmente toma bastante tiempo y dedicación.

Uso de herramientas automatizadas

- Existen muchas herramientas automatizadas que se encargan de deobfuscar el código y retornar algo más legible. Suelen ser muy fáciles de usar y, dependiendo del caso, pueden funcionar muy efectivamente.

Ejemplos:
Javascript: De4js, JS Nice
Python: deopy, Byecode Simplifier

Evaluación mixta

- Es una mezcla de la evaluación manual y la automatizada.
- Se van intercalando los pasos anteriores, “avanzando” cada vez más en la comprensión del problema.

Clase en vivo