Özgür Yazılımlar ile Statik ve Dinamik Kod Güvenliği Analizi

Emre Evren Yalçın

$Whoami

- Web Developer @ İstanbul Kültür Üniversitesi BST
- Information Security Researcher @ Signalsec.com

SignalSEC
get signal before them

Ajanda

- Statik kod analizi nedir?
- Php ile karşılaşabileceğimiz güvenlik açıkları nelerdir?
- Rips nedir? Rips ile statik kod analizi nasıl yapılır?
- Dominator nedir? Dominator ile dinamik kod analizi nasıl yapılır?
- Dünyadan bazı örnekler?
- Güvenli yazılım nasıl olmalıdır?

Statik kod analiz nedir?

- Yazılımda hata bulmak
- Yazılım kurallarına uygun bir şekilde üretim yapmak

Lexical analiz

- token_get_all, token_name
- Zend engine lexical scanner

Örnek Kullanım :

```
token_get_all("<?php echo 1; ?>");
```

Sözdizimsel ve anlamsal hatalar

- Php lint-mode

Örnek Kullanım :

```
Php -l dosyaismi
```

Kullanabileceğimiz programlar:

PHP Code Sniffer
PHP AST
PHP Sat
PHP Depend
PHP Lint
PHP Call Graph
xDebug

RIPS

- Statik kod analiz aracı
- Gelecekte php5 (wordpress, joomla)

Cross-Site Scripting (4)
SQL Injection (54)
File Disclosure (39)
File Manipulation (20)
File Inclusion (8)
Remote Code Execution(47)
Remote Command Execution (8)
Header Injection (27)
XPath Injection (3)
LDAP Injection (5)
Unserialize / POP (1)
Other (17)

Cross Site Scripting

```
<?php hello("Hello " . $_GET["name"]); ?>
```

```
/index.php?name=<script>alert(1)</script>
```

Cross Site Scripting

```
<?php hello("Hello " . $_GET["name"]); ?>
```

```
/index.php?name=<script>alert(1)</script>
```

Command İnjection

```
<?php eval("\$color = \'" . $_GET["color"] . "\';"); ?>
```

```
/index.php?color=\';phpinfo();//
```

File Inclusion

```
<?php include("includes/" . $_GET["file"]); ?>
```

```
/index.php?file=../../../../../../../etc/passwd
```

File Disclosure

```
<?php echo file_get_contents("files/" . $_GET["file"]); ?>
```

```
/index.php?file=../../../../../../../etc/passwd
```

File Manipulation

```
<?php $h = fopen($_GET["file"], "w"); fwrite($h, $_GET["data"]); ?>
```

```
/index.php?file=shell.php&data=<?php phpinfo();?>
```

Command Execution

```
<?php exec("./crypto -mode " . $_GET["mode"]); ?>
```

```
/index.php?mode=1;sleep 10;
```

Sql Injection

```
<?php mysql_query("SELECT * FROM users WHERE id = " . $_GET["id"]); ?>
```

```
/index.php?id=1 OR 1=1-- -
```

Ldap Injection

```
<?php $ctx->xpath_eval("//user[name/text()=\'" . $_GET["name"] . "\']/account/text()"); ?>
```

```
/index.php?person=*
```

POP

```
<?php
class foo {
	public $file = "test.txt";
	public $data = "text";
	function __destruct()
	{
		file_put_contents($this->file, $this->data);
	}
}
$a = unserialize($_GET["s"]);
?>
```

```
/index.php?s=O:3:"foo":2:{s:4:"file";s:9:"shell.php";s:4:"data";s:29:"
<?php passthru($_GET["c"]);?>";}
```

Grep ile güvenlik açığı keşfi

Xss :

```
grep -i -r “echo” *
grep -i -r “\$_GET” *
grep -i -r “\$_” * | grep “echo”
grep -i -r “\$_GET” * | grep “echo”
grep -i -r “\$_POST” * | grep “echo”
grep -i -r “\$_REQUEST” * | grep “echo”
```

Grep ile güvenlik açığı keşfi

Command execution:

```
grep -i -r “shell_exec(” *
grep -i -r “system(” *
grep -i -r “exec(” *
grep -i -r “popen(” *
grep -i -r “passthru(” *
grep -i -r “proc_open(” *
grep -i -r “pcntl_exec(” *
```

Grep ile güvenlik açığı keşfi

Code execution

```
grep -i -r “eval(” *
grep -i -r “assert(” *
grep -i -r “preg_replace” * | grep “/e”
grep -i -r “create_function(” *
```

Grep ile güvenlik açığı keşfi

File inclusion

```
grep -i -r “file_include” *
grep -i -r “include(” *
grep -i -r “require(” *
grep -i -r “require(\$file)” *
grep -i -r “include_once(” *
grep -i -r “require_once(” *
grep -i -r “require_once(” * | grep “\$_”
```

Grep ile güvenlik açığı keşfi

Potansiyel Sql açığı

```
grep -i -r “\$sql” *
grep -i -r “\$sql” * | grep “\$_”
```

Dinamik Analiz

- Gerçek zamanlı veri toplama
- DOM (Document Object Model) XSS
- Dominator kullanarak dom-tabanlı xss analizi gerçekleştirebilirsiniz.
- Dominator bir Firefox eklentisidir. (Firefox 3.6.13) sürümüyle çalışmaktadır.

Dominator Online

- http://www.domxssscanner.com/
- Online olarak sitelerde Dom-Xss taraması gerçekleştirebilirsiniz.

Regular Expressions

Sources

```
/(location\s*[\[.])|([.\[]\s*["']?\s*(arguments|dialogArguments|innerHTML|write(ln)?|open(Dialog)?|showModal
Dialog|cookie|URL|documentURI|baseURI|referrer
|name|opener|parent|top|content|self|frames)\W)|(localStorage|sessionStorage|Database)/
```

Sinks

```
/((src|href|data|location|code|value|action)\s*["'\]]*\s*\+?\s*=)|((replace|assign|navigate|getResponseHeader
|open(Dialog)?|showModalDialog|
eval|evaluate|execCommand|execScript|setTimeout|setInterval)\s*["'\]]*\s*\()/
```

Dominator Sources

- The Cookies Sources
- The location/documentURI/URL Sources
- The Referrer Source
- The Window Name Source
- Indirect Sources
- Other Objects Sources

Dominator Sinks

- Set Object Sinks
- Execution Sinks
- HTMLElement Sinks
- Style Sinks
- XMLHttpRequest Sink
- Set Cookie Sink
- Set Location Sink
- Control Flow Sink
- Use of Equality And Strict Equality
- Math.random Sink
- JSON Sink
- XML Sink

Neden Olduğu açıklar

- Bilgi hırsızlığı (Session)
- İçerik değişikliği
- Geçmiş tarama, port tarama
- Dahili IP çalma, Web Spidering, XSS Botnet
- Worm

Myspace Worm

- Bilinen ilk XSS wormu: 2005 yılında MySpace sosyal paylaşım sistemini etkileyen SamyJSWorm. Samy, profiline (sevdiği kitaplar benzeri bir alana) javascript yazıp gönderiyor. Bakıyor ki veritabanına kaydedildi, profil geri geldiğinde Samy'nin sayfasına, o script çalışıyor.
- Samy, profiline browserda çalıştığında (profiline baktığı anda bakan kişinin tarayıcısında) kendisini arkadaş olarak ekleyen bir kod gömüyor. Script aynı zamanda kendisini profili açan kişinin profiline de kopyalıyor ki solucan dağılsın. 24 saat içinde Samy'nin 1 milyonun üzerinde arkadaşı oluyor.

Alert('myspace');

- <script>, <body>, onclick, <a href=javascript://> taglarına izin verilmişti.
- Css tagları içinde javascript kullanılabiliyor.

Samy Worm

```
<div id="mycode" expr="alert('hah!')" style="background:url('java script:eval(document.all.mycode.expr)')">
```

”Samy is my hero”

Twitter Dom-Xss

Açık barındıran kod:

```
//<![CDATA[
(function(g){var a=location.href.split("#!")[1];if(a){g.location=g.HBR=a;}})(window);
//]]>
```

URL:

```
http://twitter.com/#!javascript:alert(document.domain);
```

Split Kullanımı

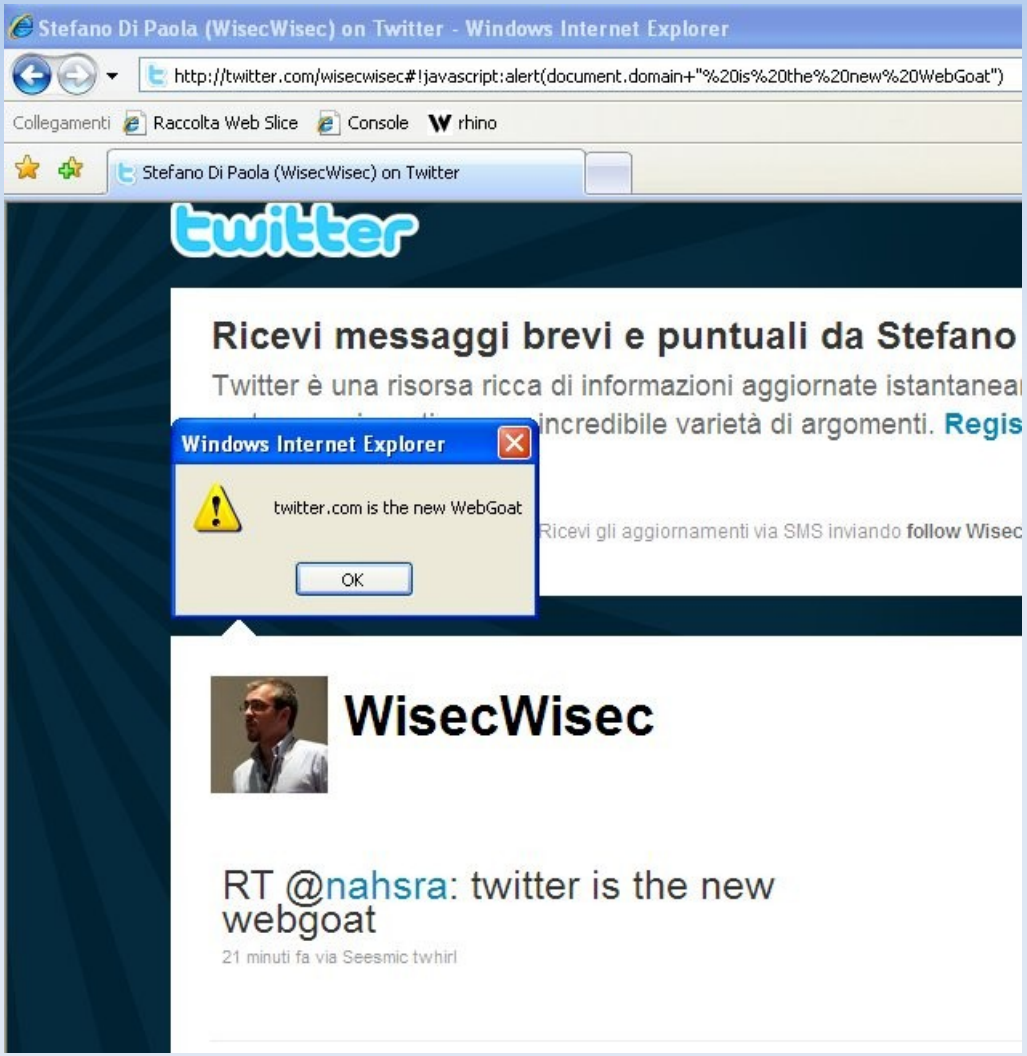
Javascript split

```
<script type="text/javascript">
var myString = "123456789";

var mySplitResult = myString.split("5");

document.write("The first element is " + mySplitResult[0]);
document.write("<br /> The second element is  " + mySplitResult[1]);
</script>
```

Sonuç:

The first element is 1234
The second element is 6789

Twitter Dom-Xss



Dom-Based Xss

```
<HTML><TITLE>Welcome!</TITLE>
Hi <SCRIPT>
var pos = document.URL.indexOf("name=") + 5;
document.write(document.URL.substring(pos,do
cument.URL.length));
</SCRIPT>
</HTML>
```

http://www.example.com/welcome.html?name=Joe

http://www.example.com/welcome.html?name=<script>alert(document.cookie)</script>

Yazılımda Güvenlik Yaklaşımı

- Statik ve dinamik kod analizi yapan otomatize araçlar riski azaltsa da, sıfır düzeyine indirememektedir.
- Güvenli mimariyi yazılım sürecinin en başında oluşturmak zaman ve maliyet açısından önemlidir.
- Yazılımda sürekli olarak iyileştirme yapılmalıdır.

Bodoslama Mimari

- Hızlı çözümler
- Heyecanlı yazılımcılar
- MVC karşıtı yaklaşımlar
- Tek bir fiziksel yapı
- Web'in büyük bir bölümü

Saldırı boyutları

- Maddi zararlar
- Manevi zararlar (prestij kaybı, negatif algı oluşması, güven kaybı)

Risk?

Advisories by SignalSEC

- Adobe Shockwave Player Memory Corruption Vulnerability
- Microsoft Windows Mobile Double Free Vulnerability
- Novell eDirector Buffer Overflow Vulnerability
- AOL 9.5 Heap Overflow Vulnerability
- Facebook (http://www.facebook.com/whitehat)
- … (yakında)

Web : http://www.signalsec.com
E-mail : evren@signalsec.com

SignalSEC
get signal before them

TEŞEKKÜRLER