

Franz Payer
Tactical Network Solutions
<http://cyberexplo.it>

EXPLOITING MUSIC STREAMING WITH JAVASCRIPT

Acknowledgements

- Zachary Cutlip
- Craig Heffner
- Tactical Network Solutions



Special Thanks

- Ronald Jenkees
 - Independent artist
 - <http://www.ronaldjenkees.com>

Legal

- ❑ EFF (www.eff.org)
- ❑ DMCA (Digital Millennium Copyright Act)
- ❑ CFAA (Computer Fraud and Abuse Act)
- ❑ Opinions/views expressed here are mine, not my employer's

What I'm going to talk about

- ❑ Background info
- ❑ Music streaming basics
- ❑ Security investigation process
- ❑ Exploit demo
- ❑ Questions

End-Goal

- ❑ Google-Chrome Extension
 - Mimics music player when possible
 - Duplicates requests otherwise
- ❑ Alternative
 - Duplicates request + caches
 - Hex-dump analysis

Wall of shame



What is streaming?

- ❑ A way to constantly receive and present data while it is being delivered by a provider – Wikipedia
- ❑ Capture data pieces
 - Reassembly
 - Encryption

What is streaming?

□ Protocols

- Custom protocol – Desktop apps
- HTTP/HTTPS – Browser apps

□ 2 types

▪ Static

- <http://cd09.128.music.static.jango.com/music/10/47/34/1047349946.mp3>

▪ Dynamic

- http://stream126-he.grooveshark.com/stream.php?streamKey=1202c0ba6260e12c0b84d64b72845181d3195496_51eaabf9_24f1b63_2cb51a8_e0616020_36_0

Types of music players

□ Flash

- Majority
- May still use JavaScript
- Must decompile
- Separate environment

□ HTML5

- Experimental
- Entirely in JavaScript
- Usually minified

Where's the vulnerability?

- ❑ Browser does heavy lifting
- ❑ Two ways to exploit
 - Copy requests
 - Easy
 - Suspicious
 - Limitations
 - Generate requests
 - Difficult
 - Undetectable w/ sessions

Investigation process






- ❑ Breadth before depth
- ❑ Locate music file in network traffic
 - Filter by XHR traffic + sort by type
- ❑ Inspect any parameters in the request
- ❑ Locate origin of those parameters
 - Page URL
 - Page source
 - localStorage
 - JavaScript
- ❑ Attempt to replicate the request

Target: Aimini









- ❑ Flash
- ❑ Almost nonexistent security
- ❑ Good first target
 - Don't even need to look at the code

Analyzing network traffic

Elements	Resources	Network	Sources	Timeline	Profiles	Audits	Console	PageSpeed
Name	Path	Method	Status	Type	Initiator	Size	Time	
			Text			Content	Latency	
	?pid=eLRJFW8CVxwrHa0905ne /view/from	GET	200 OK	text/html	Other	1.0 KB 1.5 KB	96 ms 95 ms	
	w.php?__hm=.net_View_&_lh=... www.aimini.com/webcounter	GET	200 OK	text/html	www.aimini,... Script	267 B 4 B	84 ms 84 ms	
	who_120x90_f.jpg img.aimini.net	GET	304 Not Mod	image/jpeg	www.aimini,... Parser	174 B 2.5 KB	42 ms 42 ms	
	?file=http://1.x.f.x.aimini.net/pla... 1.x.f.x.aimini.net/player/mp3	GET	200 OK	application/x-shockwave-flash	content.js:30 Script	(from c...)	25 ms 25 ms	
	?fid=XFx1jWz0zJmWApIjZdwo 1.x.f.x.aimini.net/play	GET	200 OK	audio/mp3	Other	(from c...)	185 ms 4 ms	

13 requests | 8.4 KB transferred | 1.40 s (onload: 970 ms, DOMContentLoaded: 776 ms)

The easy way out

Elements Resources Network Sources Timeline Profiles Audits Console PageSpeed								
Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency		
 ?fid=XFx1jWz0zJmWApIjZdwo /view	GET	200 OK	text/html	Other	6.1 KB 22.2 KB	415 ms 386 ms		
 ?fid=XFx1jWz0zJmWApIjZdwo 1.x.f.x.aimini.net/	GET	200	audio/mp3	Other	(from c...	185 ms 4 ms		
 ?fid=XFx1jWz0zJmWApIjZdwo 1.x.f.x.aimini.net/			image/jpeg	www.aimini... Parser	124 B 8.6 KB	184 ms 183 ms		
 ?file=http://1.x.f.x.aimini.net/			Pending	content.js:3 Script	13 B 0 B	83 ms -		
 ?file=http://1.x.f.x.aimini.net/			application/x-shockwave-flash	content.js:30 Script	(from c...)	25 ms 25 ms		
 ?nid=elBIEW8C...					1.0 KB	96 ms		

- Open link in new tab
- Copy link address
- Copy request headers
- Copy response headers
- Copy as curl

The easy way out



Downloads

Today
Jul 6, 2013



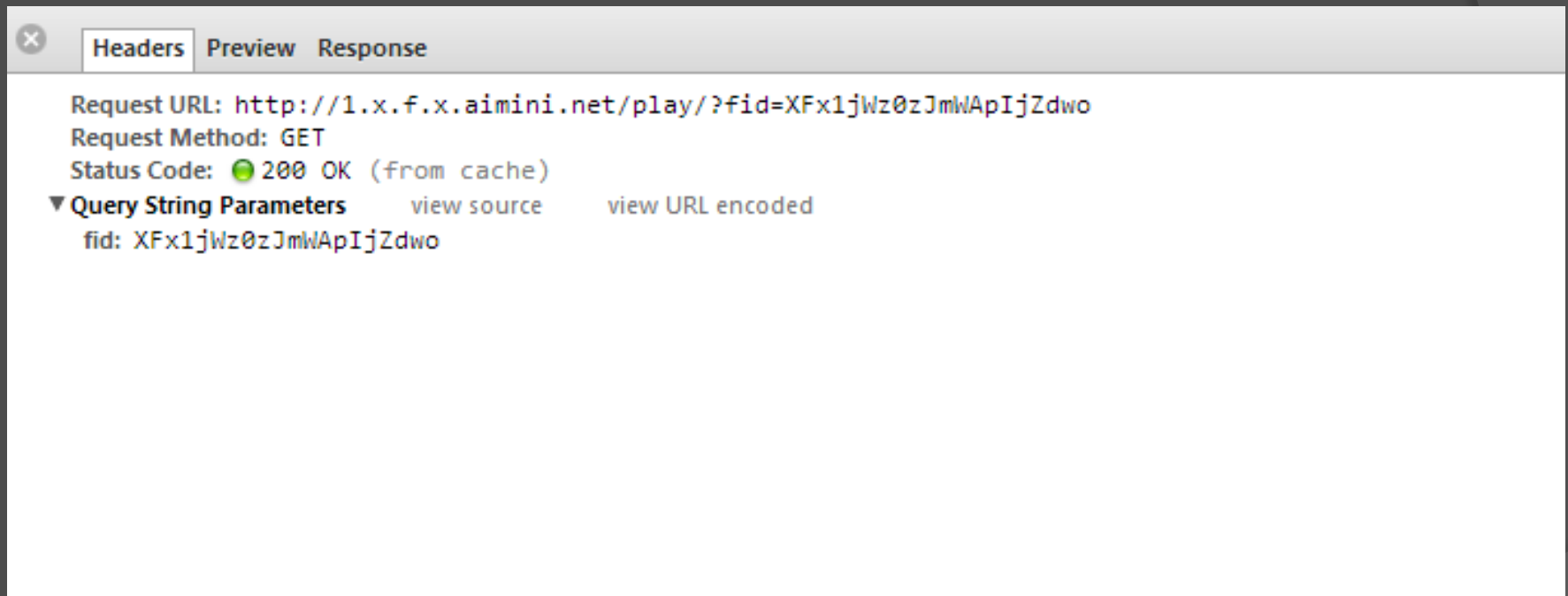
[Stay Crunchy.mp3](#)

<http://1.x.f.x.aimini.net/play/?fid=XFx1jWz0zJmWApIjZdwo>

[Show in folder](#)

[Remove from list](#)

Analyzing the song request



A screenshot of a web browser's developer tools interface, specifically the 'Headers' tab. The window title is 'Headers Preview Response'. The content shows the following details:

- Request URL: `http://1.x.f.x.aimini.net/play/?fid=XFx1jWz0zJmWApIjZdwo`
- Request Method: GET
- Status Code: ● 200 OK (from cache)
- ▼ Query String Parameters [view source](#) [view URL encoded](#)
 - fid: XFx1jWz0zJmWApIjZdwo

Looking for parameters

www.aimini.net/view/?fid=XFx1jWz0zJmWApIjZdwo

```
Request URL: http://1.x.f.x.aimini.net/play/?fid=XFx1jWz0zJmWApIjZdwo  
Request Method: GET  
Status Code: 200 OK (from cache)
```

Target: Grooveshark



- ❑ HTML5 (<http://html5.grooveshark.com/>)
- ❑ Several factors of authentication
- ❑ Minified JavaScript
- ❑ Not for the faint of heart
- ❑ Keep track of what you are doing

JavaScript beautifier

- ❑ You're going to need it
- ❑ <http://jsbeautifier.org/>

```
window.GS.tpl={"getapp.ejs":function(obj){va
'<a class="get-app" href="http://m.groovesha
" <span>"+_.getString("GET_IT_HERE")+"</span
Array.prototype.join.call(arguments,"");wit
'</h3>\n<ul class="menu">\n  <li id="nav-u
_.getString("PROFILE")+</a>\n  </li>\n
'/collection" data-translate-text="COLLECTIO
'/favorites" data-translate-text="FAVORITES"
'/playlists" data-translate-text="PLAYLISTS"
'/following" data-translate-text="FOLLOWING"
with(obj||{})__p+='<div class="banner">\n
(image.alt||"")+>\n  </a>\n  ',hasClos
{var __p="",print=function(){__p+=Array.prot
' alt="'+(image.alt||"")+>\n</a>';return
"";var style="";style+="animation-duration:
```



```
window.GS.tpl = {
  "getapp.ejs": function (obj) {
    var __p = "",
        print = function () {
            __p += Array.prototype.join.
        };
    with(obj || {}) __p += '<a class="ge
        platform: platform
    }) + " <span>" + _.getString("GET_IT
    return __p
  },
  "user_menu.ejs": function (obj) {
    var __p = "",
        print = function () {
```

Analyzing the song request

Request URL: http://stream57-he.grooveshark.com/stream.php?streamKey=c94f2fd4d8f82737e441f065312436ef3e0fb288_51d8e195_24f1b63_2cb51a8_daa87234_36_0

Request Method: GET

Status Code: ● 206 Partial Content

▼ Request Headers [view source](#)

Accept: */*

DNT: 1

Host: stream57-he.grooveshark.com

Range: bytes=0-

Referer: <http://html5.grooveshark.com/>

▼ Query String Parameters [view source](#) [view URL encoded](#)

streamKey: [c94f2fd4d8f82737e441f065312436ef3e0fb288_51d8e195_24f1b63_2cb51a8_daa87234_36_0](#)

▼ Response Headers [view source](#)

Cache-Control: no-cache, no-store, must-revalidate

Connection: close

Content-Length: 7984685

Content-Range: bytes 0-7984684/7984685

Content-Type: audio/mpeg

Analyzing more.php

Request URL: <http://html5.grooveshark.com/more.php?getStreamKeyFromSongIDEx>

Request Method: POST

Status Code: 200 OK

▼ Query String Parameters [view source](#) [view URL encoded](#)

getStreamKeyFromSongIDEx:

▼ Request Payload [view source](#)

▼ {header:{client:mobileshark, clientRevision:20120830, privacy:0,...}, method:getStreamKeyFromSongIDEx,...}

▼ header: {client:mobileshark, clientRevision:20120830, privacy:0,...}

client: "mobileshark"

clientRevision: "20120830"

▶ country: {ID:223, CC1:0, CC2:0, CC3:0, CC4:1073741824, DMA:512, IPR:0}
privacy: 0

session: "86950c0f84cc66f2e26e92b869c5d4e1"

token: "1f2ad15df0392695236c07d9ae968c3489a8a8cf9db3a6"

uuid: "38D1D238-7C51-4B5F-9EDB-F79B70DE7EE5"

method: "getStreamKeyFromSongIDEx"

▼ parameters: {prefetch:false, mobile:true, songID:38738787,...}

▶ country: {ID:223, CC1:0, CC2:0, CC3:0, CC4:1073741824, DMA:512, IPR:0}
mobile: true
prefetch: false

songID: 38738787

Analyzing more.php

Request URL: <https://html5.grooveshark.com/more.php?getCommunicationToken>

Request Method: POST

Status Code: ● 200 OK

▼ Query String Parameters [view source](#) [view URL encoded](#)

getCommunicationToken:

▼ Request Payload [view source](#)

▼ {header:{client:mobileshark, clientRevision:20120830,...}, method:getCommunicationToken,...}

▶ header: {client:mobileshark, clientRevision:20120830,...}
method: "getCommunicationToken"

▼ parameters: {secretKey:51f4d8932bdc94f2dc777e9f00a205ee}

secretKey: "51f4d8932bdc94f2dc777e9f00a205ee"

So now what?

- ❑ We need:
 - streamKey
- ❑ How do we get it?
 - more.php - getStreamKeyFromSongIDEx
 - Session - ?
 - Token - ?
 - UUID - ?
 - songID - ?
- ❑ more.php - getCommunicationToken

Looking through app.min.js

```
window.GS.tpl = {
  "getapp.ejs": function (obj) {
    var __p = "",
        print = function () {
            __p += Array.prototype.join.
        };
    with(obj || {}) __p += '<a class="ge
      platform: platform
    }) + " <span>" + _.getString("GET_IT
    return __p
  },
  "user_menu.ejs": function (obj) {
    var __p = "",
        print = function () {
```

window.GS.config

```
▼ Object {country: Object, runMode: "production",
  IP: "██████████"}
  ► country: Object
  lang: "en"
  runMode: "production"
  sessionId: "86950c0f84cc66f2e26e92b869c5d4e1"
  ► user: Object
  ► __proto__: Object
```

window.GS.models.queue.models

```
[▼ t.hasOwnProperty.i ⓘ ]
  ► _callbacks: Object
  _changed: false
  _changing: false
  ► _escapedAttributes: Object
  ► _previousAttributes: Object
  ► attributes: Object
  cid: "c30"
  ► collection: t.hasOwnProperty.i
  id: 38738787
  ► __proto__: y
```

Recap

- ❑ We need:
 - streamKey
- ❑ How do we get it?
 - more.php - getStreamKeyFromSongIDEx
 - Session – window.GS.config
 - Token - ?
 - UUID - ?
 - songID - window.GS.models.queue.models
- ❑ more.php - getCommunicationToken

Looking for variables – app.min.js

```
loaded: function () {  
    return this.length > 0 || !! this._loaded  
}  
}, _mixin({  
    UUID: function () {  
        return "xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx".replace(/[xy]/g, function (e) {  
            var t = Math.random() * 16 | 0,  
                n = e == "x" ? t : t & 3 | 8;  
            return n.toString(16)  
        }).toUpperCase()  
    },  
    getString: function (e, n) {  
        var r = $.localize.getString(e),
```

Recap

- ❑ We need:
 - `streamKey`
- ❑ How do we get it?
 - `more.php - getStreamKeyFromSongIDEx`
 - `Session – window.GS.config`
 - `Token - ?`
 - `UUID – copied function from app.min.js`
 - `songID - window.GS.models.queue.models`
- ❑ `more.php - getCommunicationToken`

Looking for variables – app.min.js

```
var p;  
r.lastRandomizer = o();  
p = hex_sha1([this.method, r.currentToken, r.revToken, r.lastRandomizer].join(":"));  
f.header.token = r.lastRandomizer + p
```

Request URL: <http://html5.grooveshark.com/more.php?getStreamKeyFromSongIDEx>

Request Method: POST

Status Code: ● 200 OK

▼ Query String Parameters [view source](#) [view URL encoded](#)

getStreamKeyFromSongIDEx:

▼ Request Payload [view source](#)

▼ {header:{client:mobleshark, clientRevision:20120830, privacy:0,...}, method:getStreamKeyFromSongIDEx,...}

▼ header: {client:mobleshark, clientRevision:20120830, privacy:0,...}

client: "mobleshark"

clientRevision: "20120830"

▶ country: {ID:223, CC1:0, CC2:0, CC3:0, CC4:1073741824, DMA:512, IPR:0}

privacy: 0

session: "86950c0f84cc66f2e26e92b869c5d4e1"

token: "1f2ad15df0392695236c07d9ae968c3489a8a8cf9db3a6"

uuid: "38D1D238-7C51-4B5F-9EDB-F79B70DE7EE5"

method: "getStreamKeyFromSongIDEx"

▼ parameters: {prefetch:false, mobile:true, songID:38738787,...}

▶ country: {ID:223, CC1:0, CC2:0, CC3:0, CC4:1073741824, DMA:512, IPR:0}

mobile: true

prefetch: false

songID: 38738787

Looking for variables – app.min.js

```
var p;  
r.lastRandomizer = o();  
p = hex_sha1([this.method, r.currentToken, r.revToken, r.lastRandomizer].join(":"));  
f.header.token = r.lastRandomizer + p
```

```
function o() {  
    var e = "";  
    for (var t = 0; t < 6; t++) e += Math.floor(Math.random() * 16).toString(16);  
    return e != r.lastRandomizer ? e : o()  
}
```

Looking for variables – app.min.js

```
var p;  
r.lastRandomizer = o();  
p = hex_sha1([this.method, r.currentToken, r.revToken, r.lastRandomizer].join(":"));  
f.header.token = r.lastRandomizer + p
```


```
    INVALID_CLIENT: 1024,  
    RATE_LIMITED: 512,  
    INVALID_TOKEN: 256,  
    INVALID_SESSION: 16,  
    MAINTENANCE: 10,  
    MUST_BE_LOGGED_IN: 8,  
    EMPTY_RESULT: -256  
  },  
  headers: {  
    client: "mobileshark",  
    clientRevision: "20120830"  
  },  
  revToken: n,
```

Looking for variables – app.min.js

```
var p;  
r.lastRandomizer = o();  
p = hex_sha1([this.method, r.currentToken, r.revToken, r.lastRandomizer].join(":"));  
f.header.token = r.lastRandomizer + p
```

Request URL: <https://html5.grooveshark.com/more.php?getCommunicationToken>

Request Method: POST

Status Code:  200 OK

▼ Query String Parameters [view source](#) [view URL encoded](#)

```
h(), r.tokenPending = !0, r.sessionID ? (e = hex_md5(r.sessionID),  
secretKey: e
```

▶ header: {client:mobileshark, clientRevision:20120830,...}
method: "getCommunicationToken"

▼ parameters: {secretKey:51f4d8932bdc94f2dc777e9f00a205ee}
secretKey: "51f4d8932bdc94f2dc777e9f00a205ee"

Recap

- ❑ We need:
 - streamKey
- ❑ How do we get it?
 - more.php - getStreamKeyFromSongIDEx
 - Session – window.GS.config
 - Token - getCommunicationToken
 - UUID – copied function from app.min.js
 - songID - window.GS.models.queue.models
- ❑ more.php - getCommunicationToken

Demo Time



Things I learned

- ❑ Downloading music is inconvenient
- ❑ Services were fairly easy to exploit
- ❑ Impossible to completely protect streaming

Things you should know

- ❑ People have bad security (shocker)
- ❑ Several services will patch their code now
- ❑ Several services won't patch their code
- ❑ The same web-traffic logging will work with some video streaming websites too.

Case Study: Last.fm

❑ Heavily secured

- Cap bandwidth to match playback speed
- One use tokens
- Users may only have 1 stream open at a time

❑ Could not exploit

- Would require large amount of time
- Hundreds of lines of obfuscated code
- Bandwidth cap prevents stealing of entire library

Mitigations

- ❑ Current technology
 - One-time use tokens
 - Encrypted streams (rtmpe)
 - Returning songs in pieces
 - Code obfuscation
- ❑ Future proofing:
 - HTML5 audio tag with DRM support
- ❑ “HTTP Live Streaming as a Secure Streaming Method” – Bobby Kania, Luke Gusukuma

References

- ❑ Browsershark
 - <https://chrome.google.com/webstore/detail/browsershark/jhbjnipjccjloncefdoknhicbnbjjaefh>
 - Or bit.ly/18UpQtb
 - <https://github.com/fpayer/browsershark>
- ❑ Blog
 - <http://cyberexplo.it/>
- ❑ HTTP Live Streaming as a Secure Streaming Method
 - <http://vtechworks.lib.vt.edu/bitstream/handle/10919/18662/Instructions%20for%20HTTP%20Live%20Streaming%20Final.pdf>
- ❑ JS Beautifier
 - <http://jsbeautifier.org/>

Contact

- Twitter: @franz780
- fpayer@tacnetsol.com

Questions?