Cross Browser Tracking Summary Part-1

一叶飘零 / 2018-12-28 09:01:00 / 浏览数 2545 技术文章 技术文章 顶(0) 踩(0)

前言

如何通过web浏览器确定设备指纹一直是一个难题，而本篇论文(Cross Browser Tracking_NDSS17)带给我们不少新的思考与突破点。
本人将结合个人理解，对该篇论文进行原理，代码等方面的详细分析。
本次解读分为4部分：
part1：文章整体结构梳理
part2：文章技术理论简单介绍与测试
part3：文章技术理论部分代码实现解读与测试
part4：文章整体架构部分代码实现解读与测试

文章概要

文章目的

通过操作系统与硬件等不同特征，无论用户使用多少种浏览器，都可以唯一跟踪用户设备。
比如：用户A无论使用了谷歌，火狐还是IE浏览器访问了我的网站，我都可以利用该论文中的技术计算出用户A的设备指纹，且无论使用哪种浏览器，设备指纹计算结果均相同

文章由来

第一代追踪技术：根据cookie进行跟踪
第二代追踪技术：根据js获取操作系统、分辨率、像素比等设备信息，进行设备跟踪，设备归并
（但此方法很难确保准确率，因为跨浏览器后会有很多参数发生改变）
第三代追踪技术：则是发现设备后面的人。通过人的习惯、人的行为等等来对人进行归并，此项技术比较复杂。

那么本文进行的研究是2.5代追踪技术，即第二代追踪技术的改进版，希望做到无论用户切换多少种浏览器，设备的跟踪都可以具有稳定性和唯一性

文章结构

第一部分

介绍了借鉴已有技术`AmIUnique`的部分：WebGL

> WebGL（全写Web Graphics Library）是一种3D绘图协议，这种绘图技术标准允许把JavaScript和OpenGL ES 2.0结合在一起，通过增加OpenGL ES 2.0的一个JavaScript绑定，WebGL可以为HTML5
> Canvas提供硬件3D加速渲染，这样Web开发人员就可以借助系统显卡来在浏览器里更流畅地展示3D场景和模型了。
> 而作者后续的新提出的十几项渲染任务，基本是使用了该项技术

第二部分

技术改进

1.屏幕分辨率改进点

- 利用屏幕宽度和高度的比率，这样即可不随屏幕的缩放级别而改变
- 新发现参数availHeight,availWidth,availLeft,availTop,screenOrientation可用于指纹特征

2.字体列表改进点

- 曾经的字体列表基于Flash插件获取字体列表，但本技术采用侧信道方式，测量字体的高度与宽度以确定字体类型(后续文章也对这一块的代码进行了详细分析，的确可以

新技术

1.原子指纹特征
通过WebGL使用精心挑选的计算机图形参数`texture,anti-aliasing,transparency`，渲染任务，然后从渲染输出中提取特征进行分析，以区分不同浏览器

2.符合指纹特征
通过WebGL使用精心挑选的计算机图形参数`light`，渲染任务，然后从渲染输出中提取特征进行分析，以区分不同浏览器

(作者为设计了十几项渲染任务，详细任务内容在后续文章中进行阐述，但这一块的结果并未应用到跨浏览器的设备指纹计算中)

第三部分

技术实现的整体架构与细节

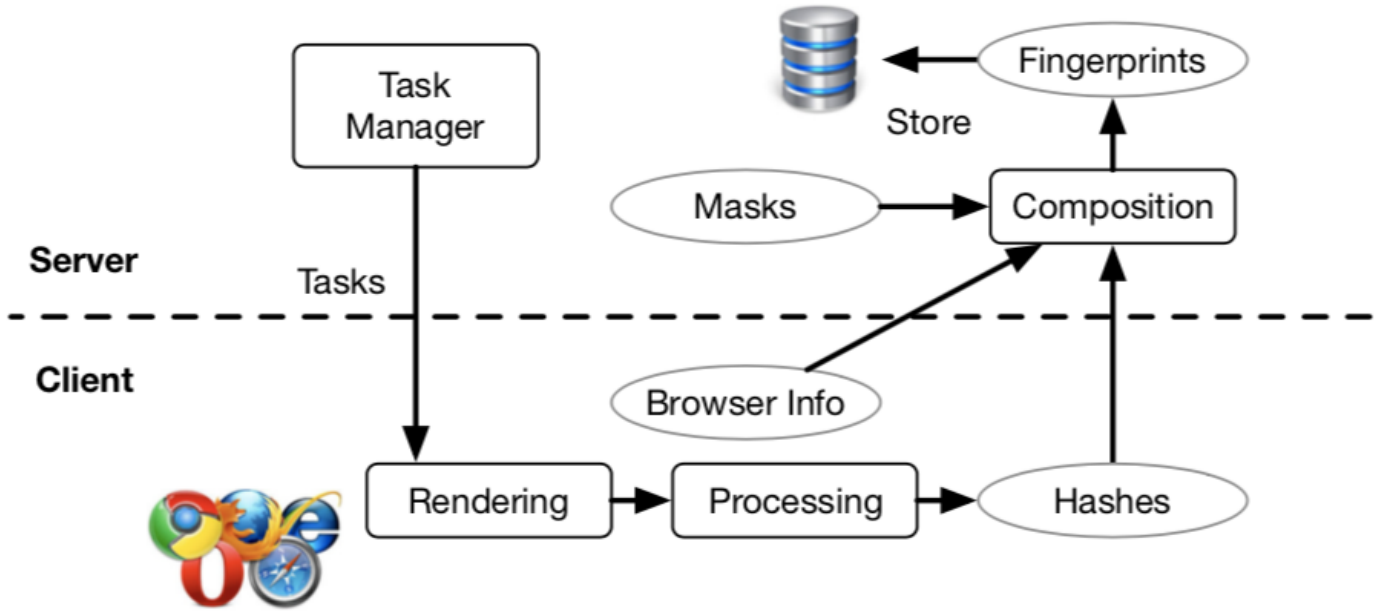

Fig. 1: System Architecture

(后续文章将结合代码分析该技术架构)

第四部分

技术的代码实现（简单介绍了代码量和使用的开源库）

> Our open-source implementation, excluding all the open-source libraries (e.g., three.js, a JavaScript 3D library, and glMatrix, a JavaScript library for matrix operations), has approximately 21K Lines of Code (LoC). Specifically, our approach involves approximately 14K lines of JavaScript, 1K lines of HTML, 2.4K lines of Coffeescript, 500 lines of C code, and 3.7K lines of Python code.

第五部分

技术的实践与数据搜集
对Amazon Mechanical Turks和MacroWorkers进行数据搜集，并和当今存在的指纹计算进行比对，得到如下结果：
计算公式

$$NH = \frac{H(X)}{H_M} = \frac{-\sum_i P(x_i) log_2 P(x_i)}{log_2(N)}$$

其中H(X)是香农熵,x是变量，值是各种概率。P(x)是概率函数。Hm是最坏情况下的熵，即每个指纹都相同，并且我们拥有最大的熵。N是所有指纹的个数。

	Ours	AmIUnique	Panopticlick
User Agent	0.612	0.570	0.531
List of Plugins	0.526	0.578	0.817
List of Fonts (Flash)	0.219	0.446	0.738
Screen Resolution	0.285	0.277	0.256
Timezone	0.340	0.201	0.161
Cookie Enabled	0.001	0.042	0.019

值得注意的是，我们发现`List of Fonts`越来越少，到本篇论文只有0.219了，这说明Flash正在逐渐被淘汰

然后又进行了唯一性和稳定性对比：

	Single-browser		Cross-browser		
	Unique	Entropy	Unique	Entropy	Stability
AmIUnique [26]	90.84%	10.82			
Boda et al. [14]			68.98%	6.88	84.64%
Ours	99.24%	10.95	83.24%	7.10	91.44%

可以看出作者的单一浏览器指纹和跨浏览器设备指纹识别率更高更稳定

第六部分

实验结果统计与分析

从浏览器分类上来看

Browser	Chrome	Firefox	Edge	IE	Opera	Safari	Other
Chrome	99.2% (100%)						
Firefox	89.1% (90.6%)	98.6% (100%)					
Edge	87.5% (92.6%)	97.9% (95.9%)	100% (100%)				
IE	85.1% (93.1%)	91.8% (90.7%)	100% (95.7%)	100% (100%)			
Opera	90.9% (90.0%)	100% (89.7%)	100% (100%)	100% (60.0%)	100% (100%)		
Safari	100% (89.7%)	100% (84.8%)	N/A	N/A	100% (100%)	100% (100%)	
Other	100% (22.2%)	100% (33.3%)	-	-	100% (50%)	-	100% (100%)

每个单元格为：唯一性（跨浏览器稳定性）

几个特点：

1.IE与Edge与其他的配对的唯一性相对低，因为其均为微软独立开发的

2.IE与Edge的稳定性很高，说明其有较多的共享代码

3.所有浏览器对Edge浏览器的唯一性都高于其对IE浏览器的唯一性，因为Edge引入更多功能，并严格遵守WebGL，暴露更多指纹信息

从特征分类来看

Feature	Single-browser	Cross-browser	
	Entropy	Entropy	Stability
User agent	6.71	0.00	1.39%
Accept	1.29	0.01	1.25%
Content encoding	0.33	0.03	87.83%
Content language	4.28	1.39	10.96%
List of plugins	5.77	0.25	1.65%
Cookies enabled	0.00	0.00	100.00%
Use of local/session storage	0.03	0.00	99.57%
Timezone	3.72	3.51	100.00%
Screen resolution and color depth	7.41	3.24	9.13%
List of fonts (Flash)	2.40	0.05	68.00%
List of HTTP headers	3.17	0.64	9.13%
Platform	2.22	1.25	97.91%
Do Not Track	0.47	0.18	82.00%
Canvas	5.71	2.73	8.17%
WebGL Vendor	2.22	0.70	16.09%
WebGL Renderer	5.70	3.92	15.39%
Use of an Ad blocker	0.67	0.28	70.78%
AmIUnique	10.82	0.00	1.39%
Screen Ratio	1.40	0.98	97.57%
List of fonts (JavaScript)	10.40	6.58	96.52%
AudioContext	1.87	1.02	97.48%
CPU Virtual cores	1.92	0.59	100.00%
Normalized WebGL Renderer	4.98	4.01	37.39%
Task (a) Texture	3.51	2.26	81.47%
Task (b) Varyings	2.59	1.76	88.25%
Task (b') Varyings+anti-aliasing	3.24	1.66	73.95%
Task (c) Camera	2.29	1.58	88.07%
Task (d) Lines&Curves	1.09	0.42	90.77%
Task (d') (d)+anti-aliasing	3.59	2.20	74.88%
Task (e) Multi-models	3.54	2.14	81.15%
Task (f) Light	3.52	2.27	81.23%
Task (g) Light&Model	3.55	2.14	80.94%
Task (h) Specular light	4.44	3.24	80.64%
Task (h') (h)+anti-aliasing	5.24	3.71	70.35%
Task (h") (h')+rotation	4.01	2.68	75.09%
Task (i) Two textures	4.04	2.68	75.98%
Task (j) Alpha (0.09)	3.41	2.36	86.25%
Task (j) Alpha (0.10)	4.11	3.02	75.31%
Task (j) Alpha (0.11)	3.95	2.84	75.80%
Task (j) Alpha (0.39)	4.35	3.06	82.75%
Task (j) Alpha (0.40)	4.38	3.10	82.58%
Task (j) Alpha (0.41)	4.49	3.13	81.89%
Task (j) Alpha (0.79)	4.74	3.12	72.63%
Task (j) Alpha (1)	4.38	3.07	82.75%
Task (k) Complex lights	6.07	4.19	66.37%
Task (k') (k)+anti-aliasing	5.79	3.96	74.45%
Task (l) Clipping plane	3.48	1.93	76.61%
Task (m) Cubemap texture	6.03	3.93	58.94%
Task (n) DDS textures	4.71	3.06	68.18%
Task (o) PVR textures	0.14	0.00	99.16%
Task (p) Float texture	5.11	3.63	74.41%
Task (q) Video	7.29	2.32	5.48%
Task (r) Writing scripts (support)	2.87	0.51	97.91%
Task (r) Writing scripts (images)	6.00	1.98	5.48%
All cross-browser features	10.92	7.10	91.44%
All features	10.95	0.00	1.39%

不难看出`List of fonts`无论是对单一浏览器还是跨浏览器都有非常重要的作用。但是由于Flash的逐渐淘汰，我们应该选择使用JavaScript来获取`List of fonts`

List of fonts (Flash)	2.40	0.05	68.00%
List of fonts (JavaScript)	10.40	6.58	96.52%

也难怪作者最后的demo，使用了该特性作为跨浏览器指纹特征之一

第七部分

简单提及了指纹识别的防御，主要是tor浏览器为例：
1.对浏览器输出做规范化处理
2.默认禁用canvas，除非用户开启
3.虚拟化，找到近乎所有可用于指纹的特征，将其虚拟化

第八、九、十部分

与本文相关技术的相关工作，本文的文献引用以及总结

后记

本篇文章主要对web指纹，设备指纹做一个简单介绍，对论文脉络做一个简单梳理。更多技术内容见后续文章。

上一篇：系统地搜索PHP disable_... 下一篇：系统地搜索PHP disable_...

1. 0 条回复
   - 动动手指，沙发就是你的了！