

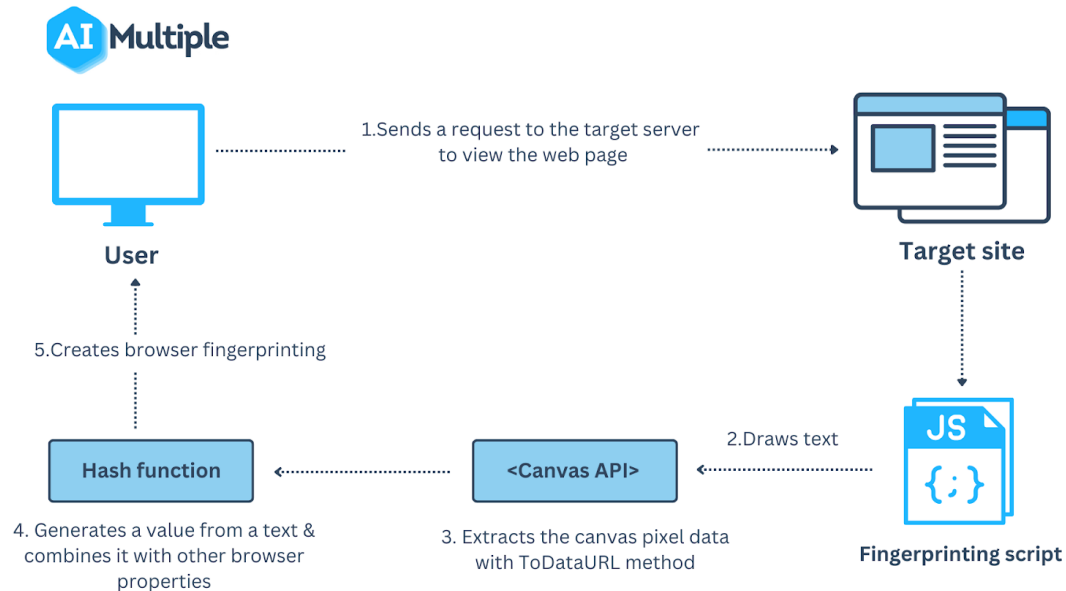
Spletne tehnologije

VAJA 2

Niko Lukač

VAJA 2

V okviru dane vaje izdelajte spletno aplikacijo za persistentno spremljanje odjemalcev, brez uporabe piškotkov (ti. **cookieless**). Namen dane vaje je, da se spoznate z osnovnimi principi delovanja modernih spletnih sledilnikov (angl. web trackers), ki so vseprisotni na svetovnem spletu, ter pri tem ugotovite načine zaščite.



Zahteve

1. Frontend:

Najprej na odjemalčevem delu izdelajte aplikacijo v JavaScriptu-u, ki naj omogoča sledeče:

- Vsaj **5 znanih tehnik identifikacije** odjemalca (pri tem naj bosta vključeni: HTML5 canvas fingerprinting ter Font fingerprinting). Vsaka tehnika naj bo implementirana v svoji funkciji/metodi. Uporaba knjižnic ni dovoljena. (3%)
- Izračun zgoščevalne vrednosti iz konkatencije rezultatov uporabljenih tehnik za identifikacijo odjemalca.
 - <https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/digest>
- Omogočite pošiljanje dane vrednosti do vaše strežniške aplikacije preko zahtevka GET. (2%)

Zahteve

- Vsaj 5 znanih tehnik identifikacije odjemalca (pri tem naj bosta vključeni: HTML5 canvas fingerprinting ter Font fingerprinting). Vsaka tehnika naj bo implementirana v svoji funkciji/metodi. Uporaba knjižnic ni dovoljena. (3%)

```
function get_text_dimensions(font) {  
    h = document.getElementsByTagName("BODY")[0];  
    d = document.createElement("DIV");  
    s = document.createElement("SPAN");  
  
    d.appendChild(s);  
    d.style.fontFamily = font;  
    s.style.fontFamily = font;  
    s.style.fontSize = "72px";  
    s.innerHTML = "font_detection";  
    h.appendChild(d);  
  
    textWidth = s.offsetWidth;  
    textHeight = s.offsetHeight;  
    h.removeChild(d);  
  
    return [textWidth, textHeight];  
}
```

```
getCanvasFingerprint: function () {  
    var canvas = document.createElement('canvas');  
    var ctx = canvas.getContext('2d');  
    var txt = 'CANVAS_FINGERPRINT';  
    ctx.font = "14px 'Arial'";  
    ctx.textBaseline = "alphabetic";  
    ctx.fillRect(125,1,62,20);  
    ctx.fillStyle = "rgba(102, 204, 0, 0.7)";  
    ctx.fillText(txt, 4, 17);  
    return canvas.toDataURL();  
}
```

Zahteve

2. Backend:

Prav tako izdelajte strežniško aplikacijo v poljubni tehnologiji (npr. node.js), ki omogoča sledeče:

- Pridobitev GET zahtevka z zgoščevalno vrednostjo s strani odjemalčeve JS aplikacije. Zgoščevalno vrednost hranite v lokalno datotečno-vodeno podatkovno bazo (npr. sqlite). V primeru, da je identična zgoščevalna vrednost že hranjena, vrnete tekstovni odziv 'true', sicer 'false'. (2%)
 - <https://expressjs.com/en/guide/routing.html>
 - <https://www.sqlitetutorial.net/sqlite-nodejs/>

Zahteve

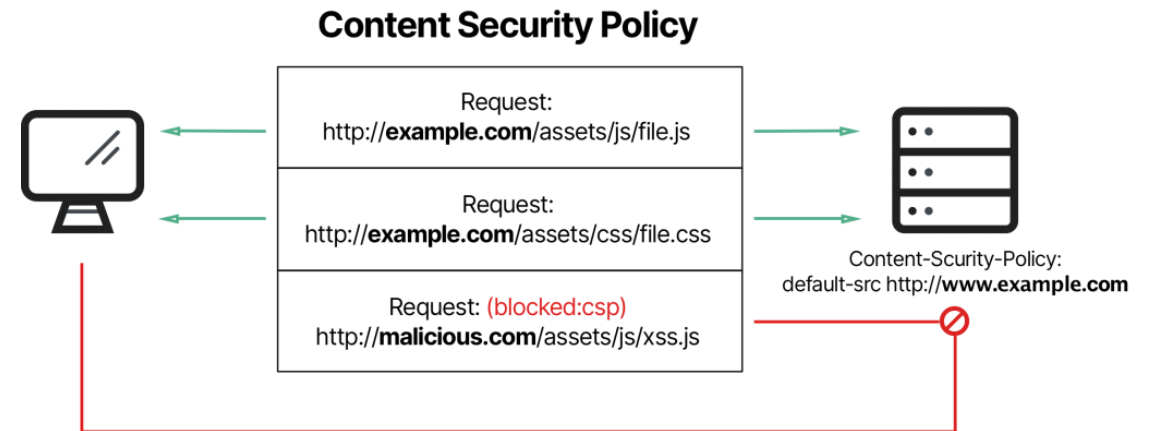
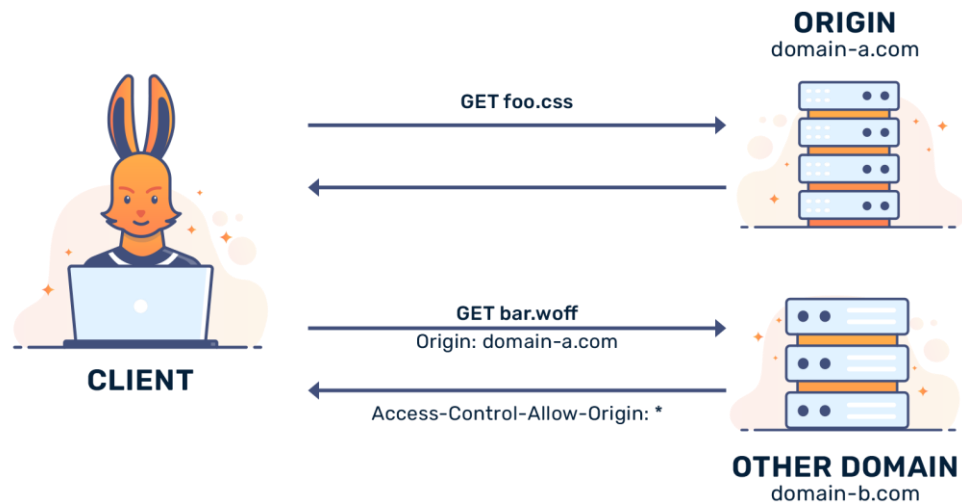
3. Testiranje in poročanje

- Delovanje demonstrirajte lokalno na preprosti spletni strani (**gostovana na ločenem strežniku, npr. nginx**), ki ob ponovnem obisku izpiše ali ste že spletno stran obiskali ali niste (tudi po ponovnem odprtju brskalnika).
- Testirajte persistentnost implementiranih tehnik identifikacije pri različni konfiguraciji spletnega brskalnika: **incognito način in uporaba vsaj enega varnostnega vtičnika**. Testirajte vpliv **CORS in CSP**.
- Na koncu podate vaše ugotovitve v obliki poročila (1-2 strani). (5%)

Zahteve

3. Testiranje in poročanje

- **CORS in CSP**



Zahteve

- **Oddajte naslednje v arhivu ZIP:** spletna aplikacija (npr. client.js in server.js), porocilo.pdf
- **Vrednost naloge:** 12% od celotne ocene pri predmetu
- **Naknadni roki za oddajo:** 2 tedna pred vsakim izpitnim rokom (datumi bodo objavljeni sproti na uvodni strani predmeta)
- Oddana vaja bo ocenjena najkasneje v 2 tednih po roku za oddajo ali v 1 tednu pred izpitnim rokom. Ustnega zagovora pri dani vaji ni. V primeru nestrinjanja z dodeljeno oceno je možen ustni zagovor po predhodnem dogovoru.