Isolated and Exhausted: Attacking Operating Systems via Site Isolation in the Browser

32nd USENIX Security Symposium (2023)

Matthias Gierlings, Marcus Brinkmann, Jörg Schwenk

2023-08-11

Chair for Network and Data Security - Ruhr University Bochum

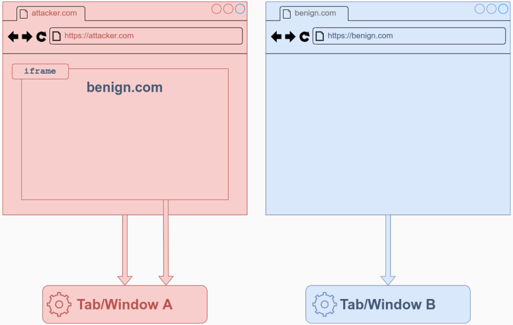
Isolated and Exhausted - Exploiting a browser security feature for attacks

How to use Site Isolation[1] to:

- DoS your browser
- DoS your OS
- Poison OS-DNS caches in the web-attacker model

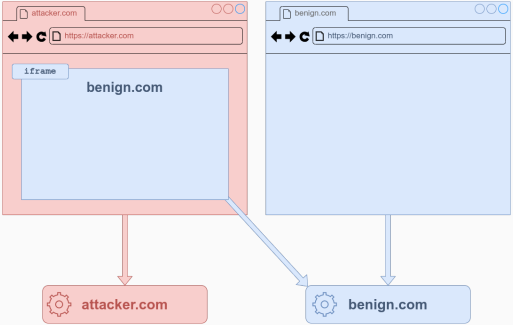
Process-Per-Tab vs. Site Isolation

Process-Per-Tab Model



- Processes shared cross site.

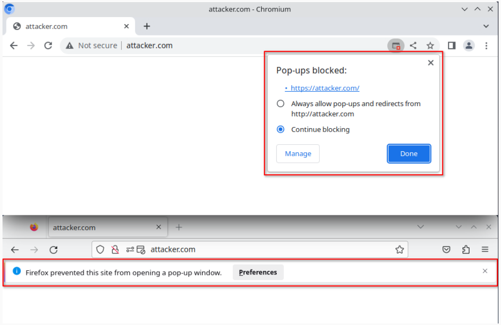
Site Isolation



- One process per site.

Process-Per-Tab vs. Site Isolation

Process-Per-Tab Model



Pop-ups blocked:

- https://attacker.com/

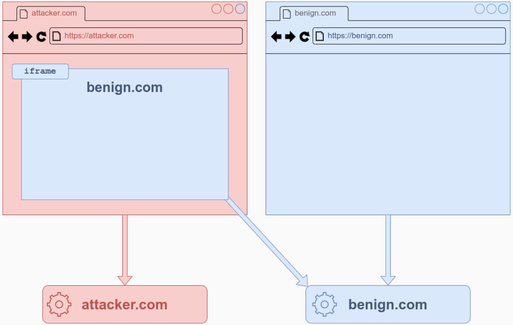
○ Always allow pop-ups and redirects from http://attacker.com

● Continue blocking

Firefox prevented this site from opening a pop-up window.

- Processes shared cross site
- Process creation **requires user interaction**

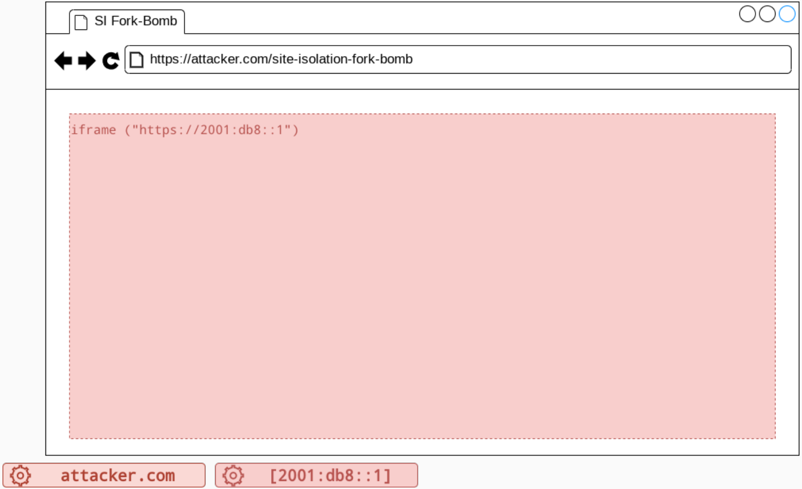
Site Isolation



- One process per site
- Process creation **automatic**

Site Isolation provides attackers with the power of process creation.

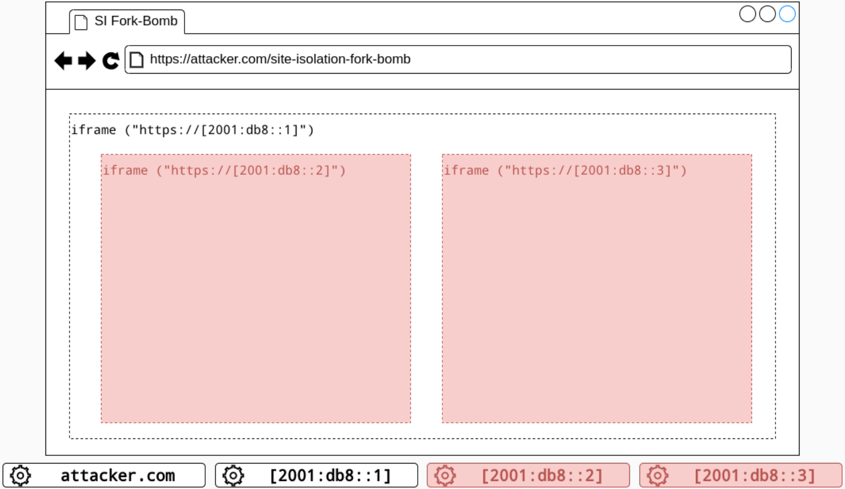
Site Isolation automatically creates processes



https://attacker.com/site-isolation-fork-bomb

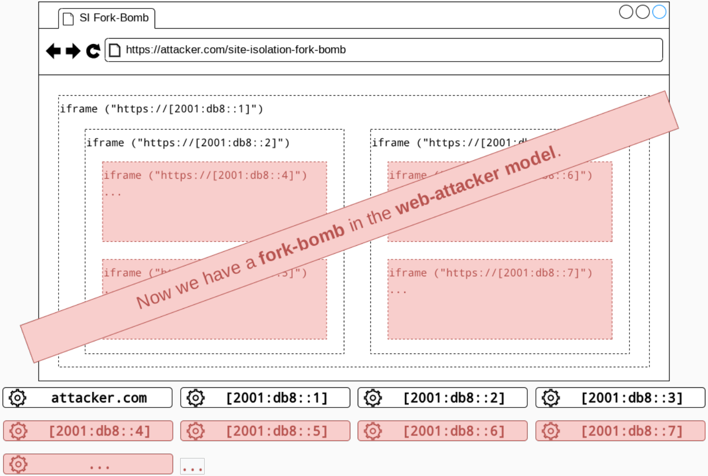
```
iframe ("https://2001:db8::1")
```

Site Isolation automatically creates processes



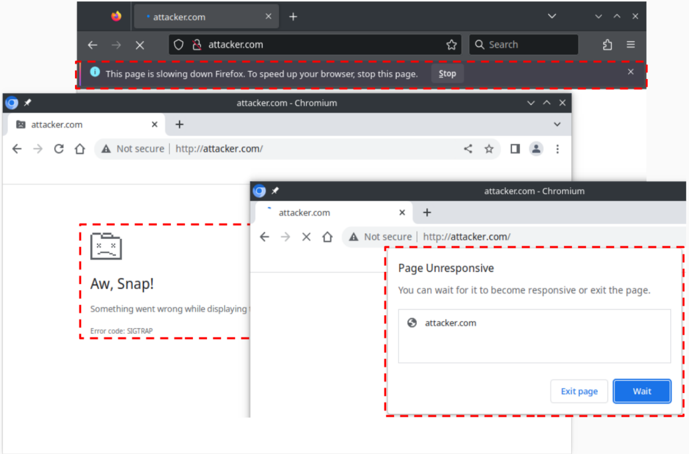
SI Fork-Bomb
https://attacker.com/site-isolation-fork-bomb
iframe ("https://[2001:db8::1]")
iframe ("https://[2001:db8::2]")
iframe ("https://[2001:db8::3]")
attacker.com
[2001:db8::1]
[2001:db8::2]
[2001:db8::3]

Site Isolation enables a fork bomb in the web-attacker model



Now we have a **fork-bomb** in the **web-attacker model**.

Resource consumption of websites is monitored and limited by browsers



This page is slowing down Firefox. To speed up your browser, stop this page.

Aw, Snap!

Something went wrong while displaying

Error code: SIGTRAP

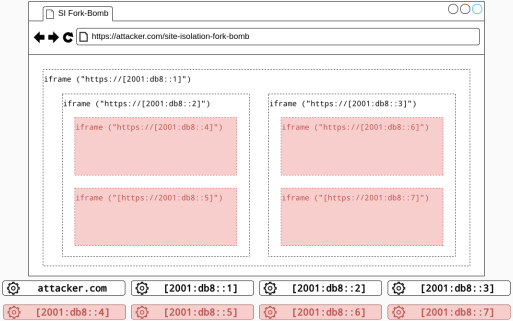
Page Unresponsive

You can wait for it to become responsive or exit the page.

attacker.com

Why does the browser fail to detect and prevent the Site Isolation fork bomb?

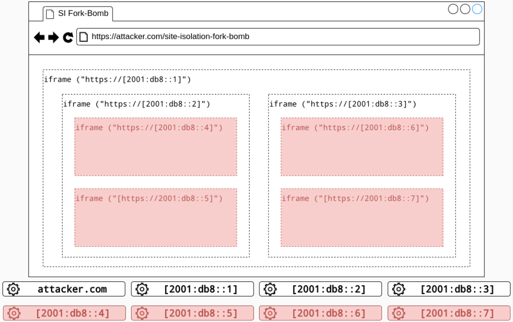
Preventing the Site Isolation fork bomb is hard


SI Fork-Bomb
https://attacker.com/site-isolation-fork-bomb
iframe ("https://[2001:db8::1]")
iframe ("https://[2001:db8::2]")
iframe ("https://[2001:db8::3]")
iframe ("https://[2001:db8::4]")
iframe ("https://[2001:db8::6]")
iframe ("[https://2001:db8::5]")
iframe ("[https://2001:db8::7]")
attacker.com
[2001:db8::1]
[2001:db8::2]
[2001:db8::3]
[2001:db8::4]
[2001:db8::5]
[2001:db8::6]
[2001:db8::7]

Attribution problem

- No information who operates a site
- *Every* site must be isolated.

Preventing the Site Isolation fork bomb is hard


SI Fork-Bomb
https://attacker.com/site-isolation-fork-bomb
iframe ("https://[2001:db8::1]")
iframe ("https://[2001:db8::2]")
iframe ("https://[2001:db8::3]")
iframe ("https://[2001:db8::4]")
iframe ("[https://2001:db8::5]")
iframe ("https://[2001:db8::6]")
iframe ("[https://2001:db8::7]")
attacker.com
[2001:db8::1]
[2001:db8::2]
[2001:db8::3]
[2001:db8::4]
[2001:db8::5]
[2001:db8::6]
[2001:db8::7]

Attribution problem

- No information who operates a site
- *Every* site must be isolated.

Monitoring problem

- Sandboxed web content barely consumes any resources
- Exhaustion caused by browser (Site Isolation overhead)

Site Isolation provides access to secondary resources

Site Isolation provides access to secondary resources

Excerpt from the Chromium source code [2]

```
// Experimentation shows that creating too many sockets creates odd problems
// because of resource exhaustion in the Unix sockets domain.
// Trouble has been seen on Linux at 3479 sockets in test, so leave a margin.
const int kMaxSimultaneousSockets = 3000;
```

Site Isolation provides access to secondary resources

Excerpt from the Chromium source code [2]

```
// Experimentation shows that creating too many sockets creates odd problems
// because of resource exhaustion in the Unix sockets domain.
// Trouble has been seen on Linux at 3479 sockets in test, so leave a margin.
const int kMaxSimultaneousSockets = 3000;
```

Potential problem: A global shared limit enables **DoS via single site**.

Site Isolation provides access to secondary resources

Excerpt from the Chromium source code [2]

```
// Experimentation shows that creating too many sockets creates odd problems
// because of resource exhaustion in the Unix sockets domain.
// Trouble has been seen on Linux at 3479 sockets in test, so leave a margin.
const int kMaxSimultaneousSockets = 3000;
```

Potential problem: A global shared limit enables **DoS via single site**.

Chromium limits sockets per-process.

Process-per-tab model

One window/tab **can not DoS the browser**

Site Isolation provides access to secondary resources

Excerpt from the Chromium source code [2]

```
// Experimentation shows that creating too many sockets creates odd problems
// because of resource exhaustion in the Unix sockets domain.
// Trouble has been seen on Linux at 3479 sockets in test, so leave a margin.
const int kMaxSimultaneousSockets = 3000;
```

Potential problem: A global shared limit enables **DoS via single site**.
Chromium limits sockets per-process.

Process-per-tab model

One window/tab **can not DoS the browser**.

Site Isolation

One window/tab **can DoS the entire OS**.

DNS Cache Poisoning in the web-attacker model

- **DNS-Poisoning** attack against a Windows 10 client . . .
  - in $\approx$ 3.5 min median time . . .
  - fastest attack iteration: 15 s
- **Exhausting** the UDP ephemeral port pool, bypassing resource limits via Site Isolation
  - The victim must use fixed DNS-Query port instead of a random one
- **Misappropriating** WebRTC to create many idle **network sockets**.

DNS Cache Poisoning in the web-attacker model

DNS-Poisoning by
Exhaustive
Misappropriation
Of
Network
Sockets

DNS Cache Poisoning in the web-attacker model

DNS-Poisoning by

Exhaustive

Misappropriation

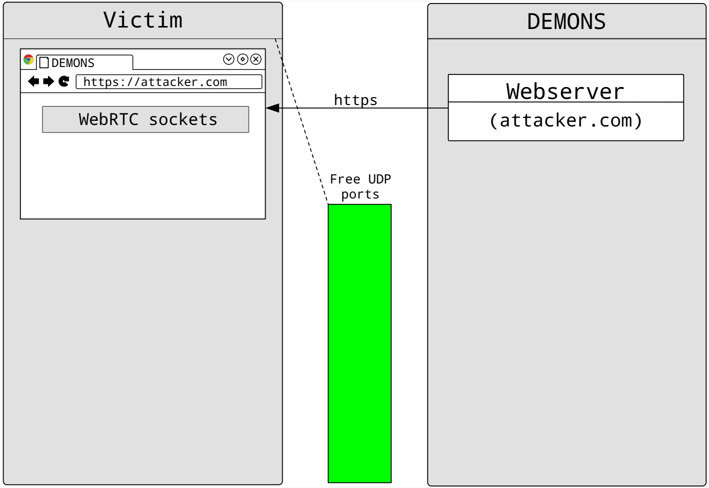
Of

Network

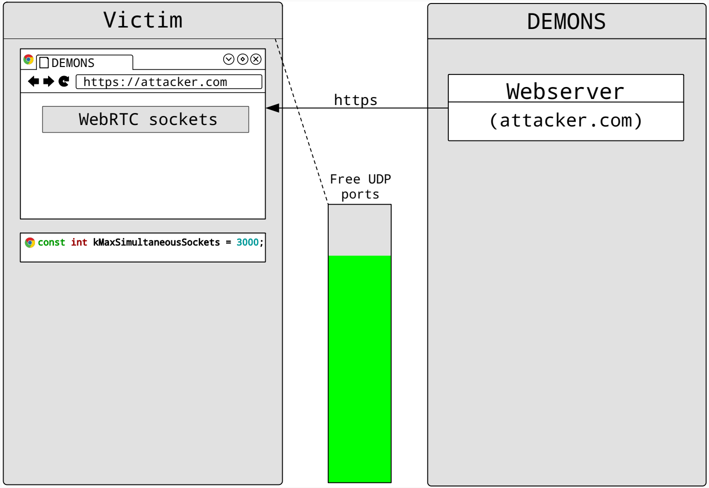
Sockets

DEMONS in a nutshell

DEMONS Setup Phase - Exhaustion of the ephemeral port pool

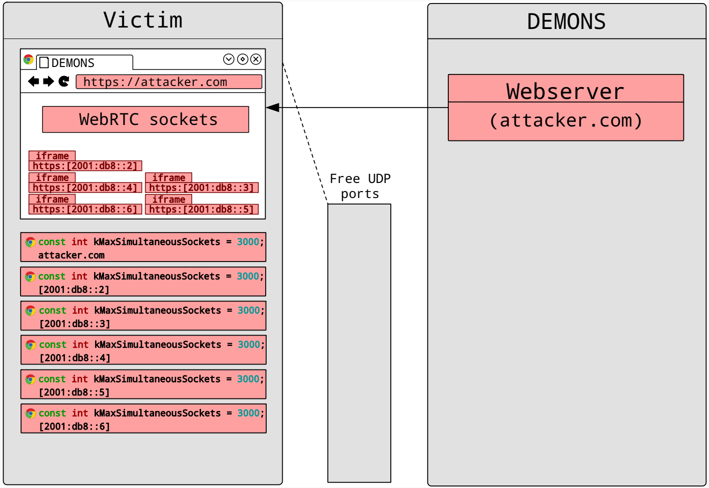


DEMONS Setup Phase - Exhaustion of the ephemeral port pool



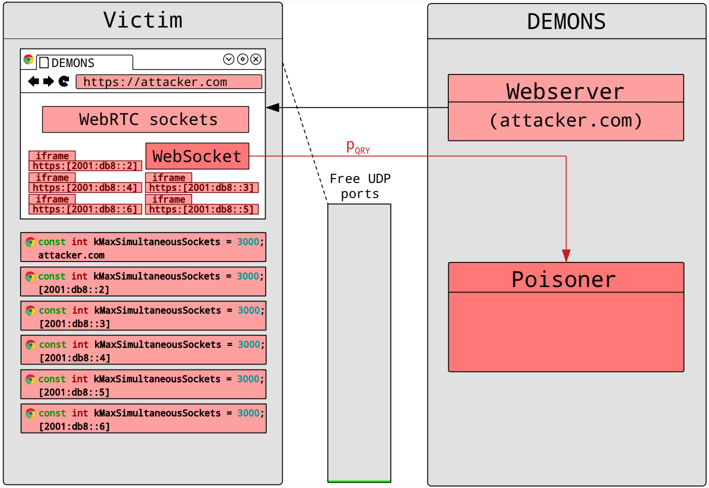
```
const int kMaxSimultaneousSockets = 3000;
```

DEMONS Setup Phase - Exhaustion of the ephemeral port pool



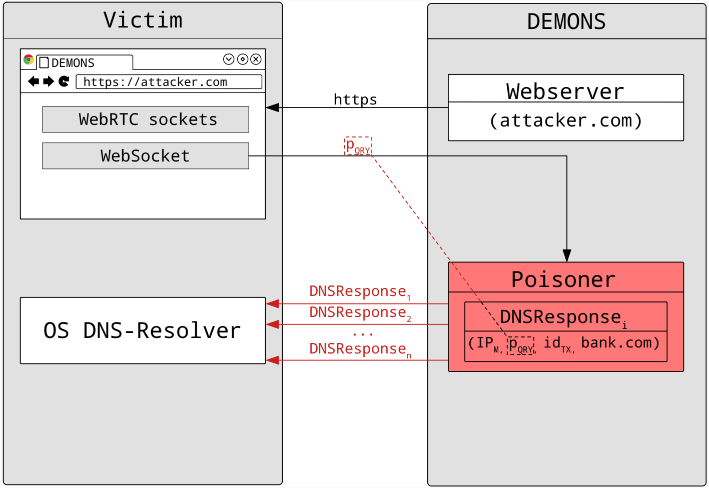
```
const int kMaxSimultaneousSockets = 3000;
attacker.com
const int kMaxSimultaneousSockets = 3000;
[2001:db8::2]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::3]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::4]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::5]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::6]
```

DEMONS Setup Phase - Leaking the DNS query port

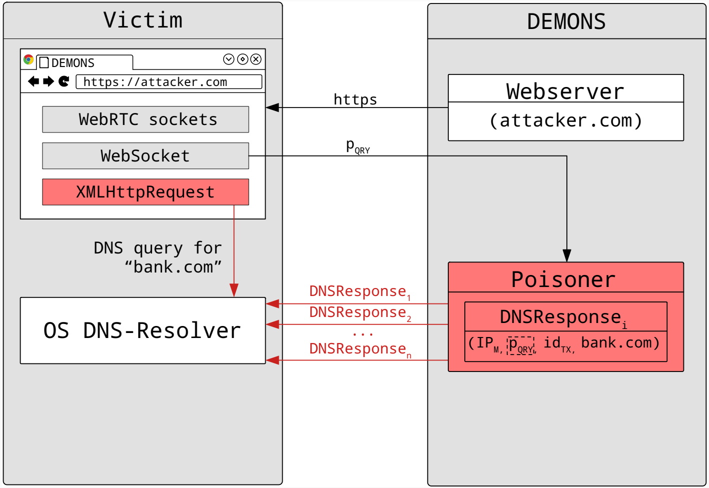


```
const int kMaxSimultaneousSockets = 3000;
attacker.com
const int kMaxSimultaneousSockets = 3000;
[2001:db8::2]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::3]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::4]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::5]
const int kMaxSimultaneousSockets = 3000;
[2001:db8::6]
```

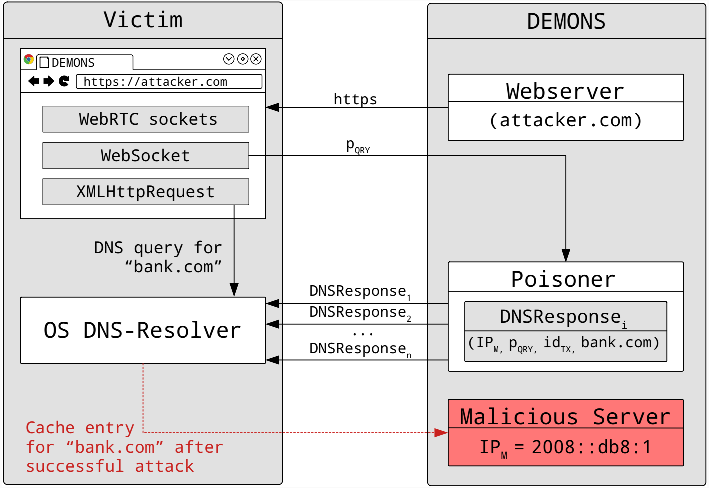
DEMONS Poisoning Phase - Sending spoofed responses



DEMONS Poisoning Phase - Triggering the DNS query



DEMONS - State after a successful attack



The current state of Site Isolation

DEMONS (CVE-2020-6557) is mitigated via global port limit.

Drawbacks:

- May not suffice if multiple browsers are in use
- Global limit can enable DoS against browser

Site Isolation fork bomb is currently not mitigated.

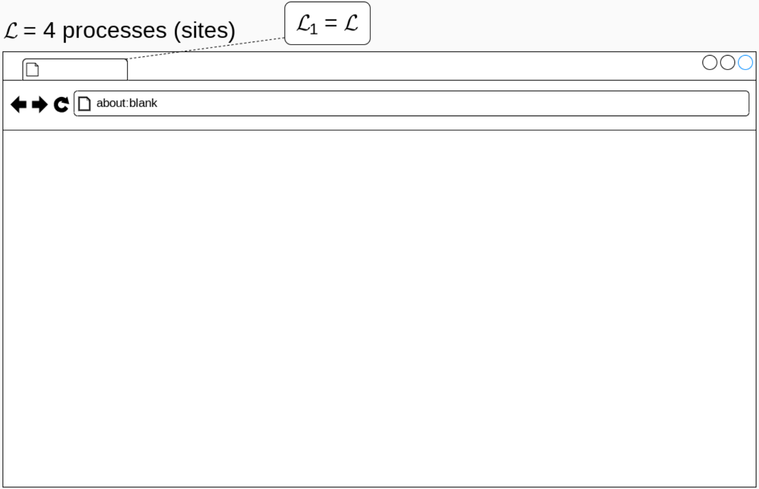
Countermeasure[1]/PoC[2] proposed.

[1] https://bugzilla.mozilla.org/show_bug.cgi?id=1722160, https://bugs.chromium.org/p/chromium/issues/detail?id=1094876

[2] Chromium only, submitted via bug tracker in August 2022

Fixing the Site Isolation fork bomb

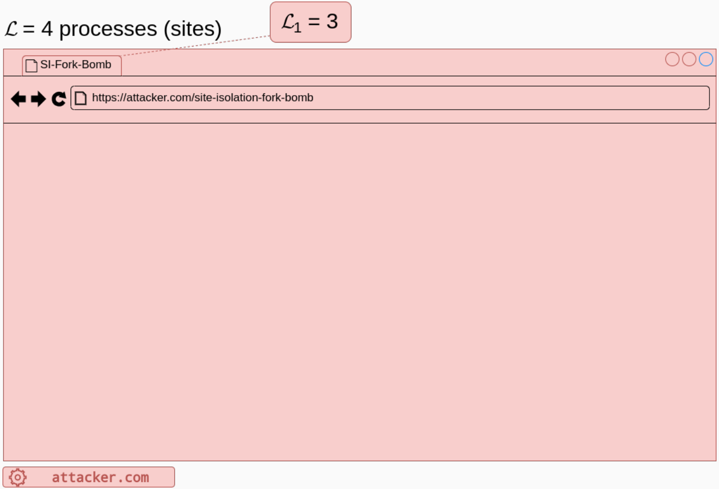
$\mathcal{L}$ = 4 processes (sites)



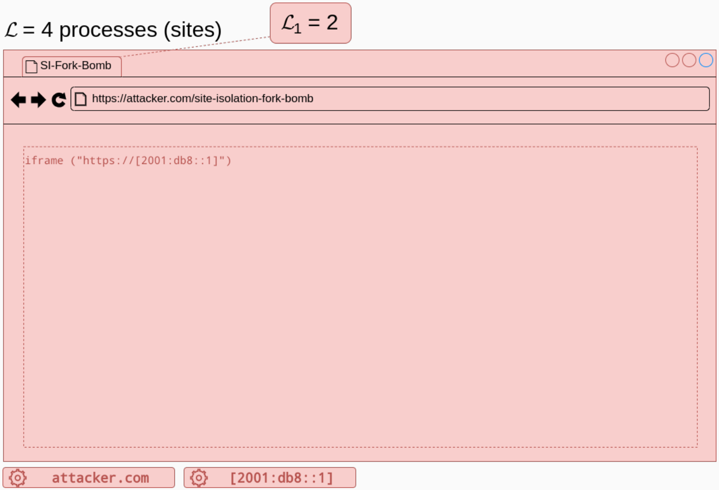
$\mathcal{L}_1 = \mathcal{L}$

Fixing the Site Isolation fork bomb

$\mathcal{L}$ = 4 processes (sites)



Fixing the Site Isolation fork bomb



$\mathcal{L}$ = 4 processes (sites)

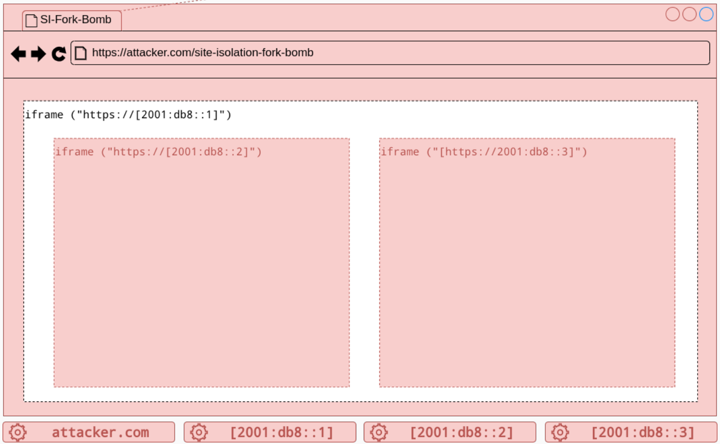
$\mathcal{L}_1 = 2$

```
iframe ("https://[2001:db8::1]")
```

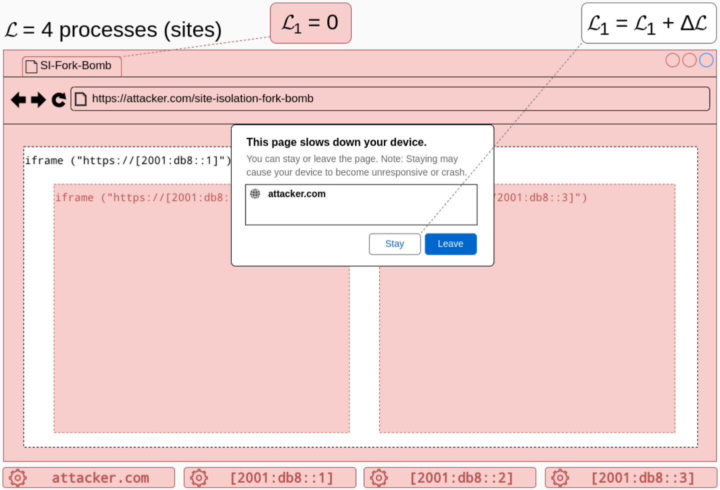
Fixing the Site Isolation fork bomb

$\mathcal{L}$ = 4 processes (sites)

$\mathcal{L}_1 = 0$


SI-Fork-Bomb
https://attacker.com/site-isolation-fork-bomb
iframe ("https://[2001:db8::1]")
iframe ("https://[2001:db8::2]")
iframe ("[https://2001:db8::3]")
attacker.com
[2001:db8::1]
[2001:db8::2]
[2001:db8::3]

Fixing the Site Isolation fork bomb

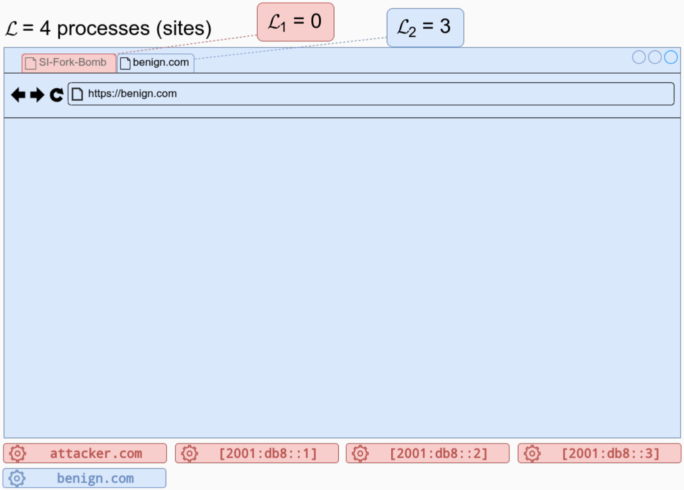


$\mathcal{L}$ = 4 processes (sites)
$\mathcal{L}_1 = 0$
$\mathcal{L}_1 = \mathcal{L}_1 + \Delta\mathcal{L}$
SI-Fork-Bomb
https://attacker.com/site-isolation-fork-bomb
iframe ("https://[2001:db8::1]")
iframe ("https://[2001:db8:
2001:db8::3]")
attacker.com
Stay
Leave
attacker.com
[2001:db8::1]
[2001:db8::2]
[2001:db8::3]

This page slows down your device.

You can stay or leave the page. Note: Staying may cause your device to become unresponsive or crash.

Fixing the Site Isolation fork bomb

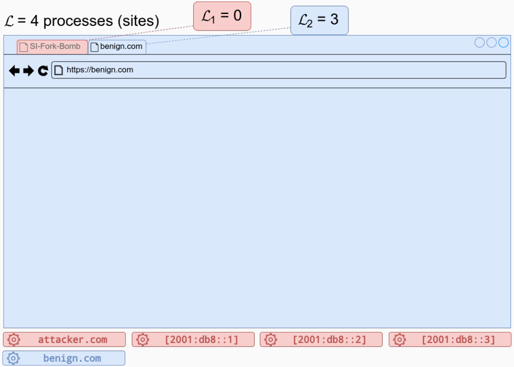


$\mathcal{L}$ = 4 processes (sites)

$\mathcal{L}_1 = 0$

$\mathcal{L}_2 = 3$

Fixing the Site Isolation fork bomb



PoC Mitigation with $\mathcal{L} = \Delta\mathcal{L} = 30$

- Prevents fork bomb
- Prevents DoS on browser
- Unlikely to affect user experience (tested against Tranco[3] Top 1000)
- Can utilize existing notification mechanisms

Resources & Contact

Artifacts available[a]

DOI 10.5281/zenodo.7356538

Caution save your work before you try the fork bomb.

DEMONS Demo Video

via Chomium Bug 1083278[b]

Contact

Matthias Gierlings
(matthias.gierlings@rub.de)

Questions?

[a] Zenodo: https://doi.org/10.5281/zenodo.7356538, GitLab Mirror: https://git.noc.ruhr-uni-bochum.de/gierlmds/isolated-and-exhausted

[b] https://bugs.chromium.org/p/chromium/issues/detail?id=1083278

References i

[1] C. Reis, A. Moshchuk, and N. Oskov, “Site Isolation: Process Separation for Web Sites within the Browser,” in *28th USENIX security symposium (USENIX security 19)*, 2019, pp. 1661–1678 [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/reis

[2] The Chromium Authors, 2023. [Online]. Available: https://github.com/chromium/chromium/blob/fd8a8914ca0183f0add65ae55f04e287543c7d4a/services/network/p2p/socket_manager.cc#L45%0A

[3] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” in *Proceedings of the 26th annual network and distributed system security symposium*, 2019, doi: 10.14722/ndss.2019.23386.