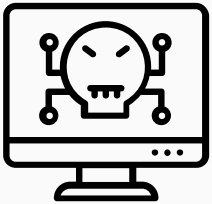
JavaScript Template Attacks

Michael Schwarz, Florian Lackner, Daniel Gruss

Februrary 25, 2019

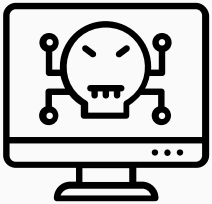
IAIK – Graz University of Technology

Motivation



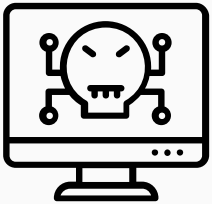
- Many (undocumented) properties in JavaScript sandboxes

Motivation



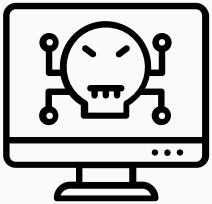
- Many (undocumented) properties in JavaScript sandboxes
- Properties should not leak environment info

Motivation

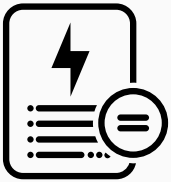


- Many (undocumented) properties in JavaScript sandboxes
- Properties should not leak environment info
- Information useful for exploits and side-channel attacks

Motivation

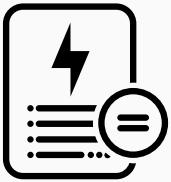


- Many (undocumented) properties in JavaScript sandboxes
- Properties should not leak environment info
- Information useful for exploits and side-channel attacks
- Also usable for fingerprinting

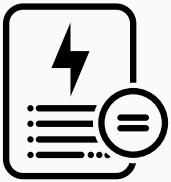


- Theory: JavaScript sandbox is environment agnostic

Identifying Properties

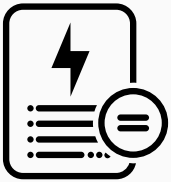


- Theory: JavaScript sandbox is environment agnostic
- Code gives same results independent of platforms or hardware

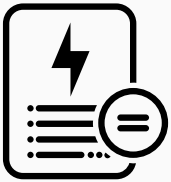


- Theory: JavaScript sandbox is environment agnostic
- Code gives same results independent of platforms or hardware
- `window.Array.name` is always "`Array`"

Identifying Properties

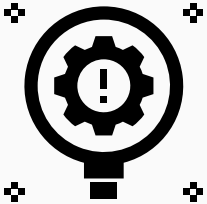


- Theory: JavaScript sandbox is environment agnostic
- Code gives same results independent of platforms or hardware
- `window.Array.name` is always "`Array`"
- Some defined exceptions, e.g., user agent



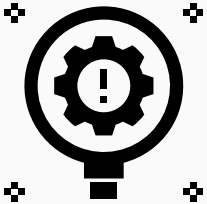
- Theory: JavaScript sandbox is environment agnostic
- Code gives same results independent of platforms or hardware
- `window.Array.name` is always "`Array`"
- Some defined exceptions, e.g., user agent
- Tor browser → identifying properties anonymized

Information Leakage



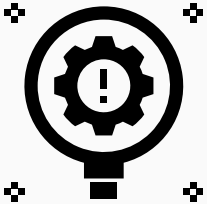
- Properties leaking info about hardware or software...

Information Leakage



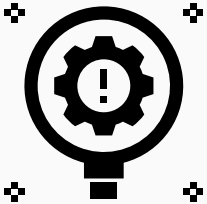
- Properties leaking info about hardware or software...
  - ...can be used to track users ($\rightarrow$ fingerprinting)

Information Leakage

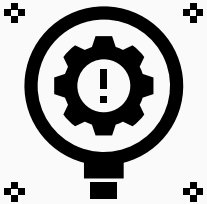


- Properties leaking info about hardware or software...
  - ...can be used to track users ($\rightarrow$ fingerprinting)
  - ...make phishing more plausible

Information Leakage

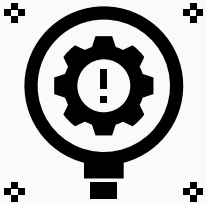


- Properties leaking info about hardware or software...
  - ...can be used to track users ($\rightarrow$ fingerprinting)
  - ...make phishing more plausible
  - ...allow selecting fitting exploits



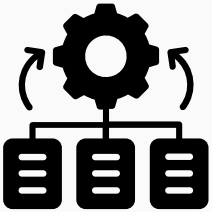
- Properties leaking info about hardware or software...
  - ...can be used to track users ($\rightarrow$ fingerprinting)
  - ...make phishing more plausible
  - ...allow selecting fitting exploits
  - ...provide necessary information for side-channel attacks

Information Leakage



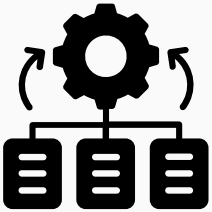
- Properties leaking info about hardware or software...
  - ...can be used to track users ($\rightarrow$ fingerprinting)
  - ...make phishing more plausible
  - ...allow selecting fitting exploits
  - ...provide necessary information for side-channel attacks
- $\rightarrow$ indirect security risk

Automated Leakage Identification



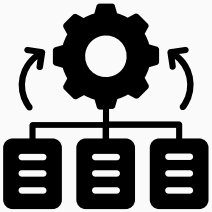
- Manually finding leakage → time consuming

Automated Leakage Identification



- Manually finding leakage → time consuming
- Automate the task

Automated Leakage Identification

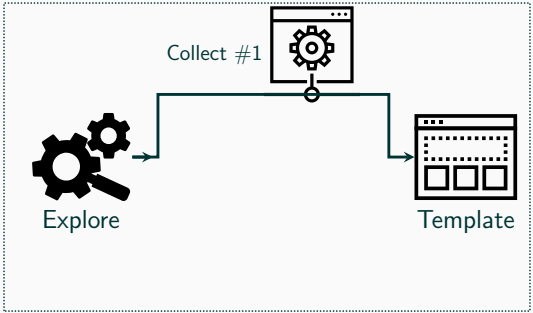


- Manually finding leakage → time consuming
- Automate the task
- Idea of template attacks: change a factor, compare results

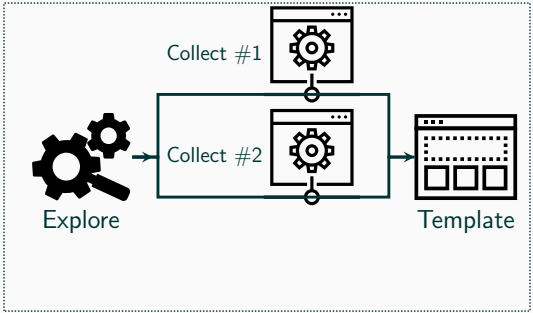

JS
Template Attacks

Explore

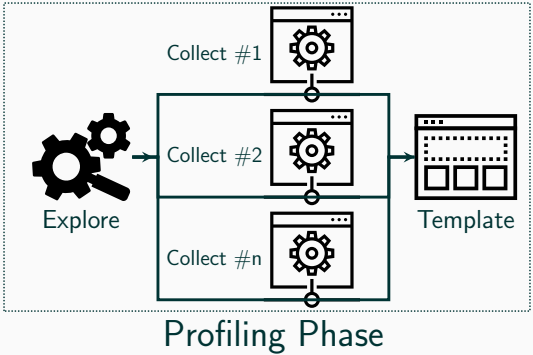
Profiling Phase



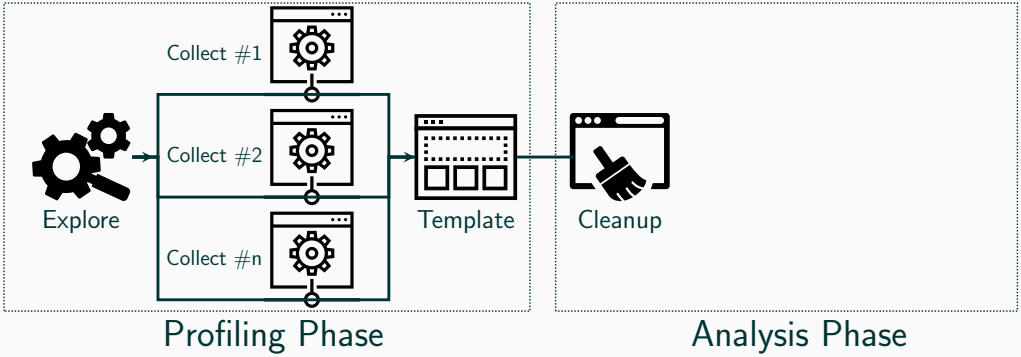
Profiling Phase



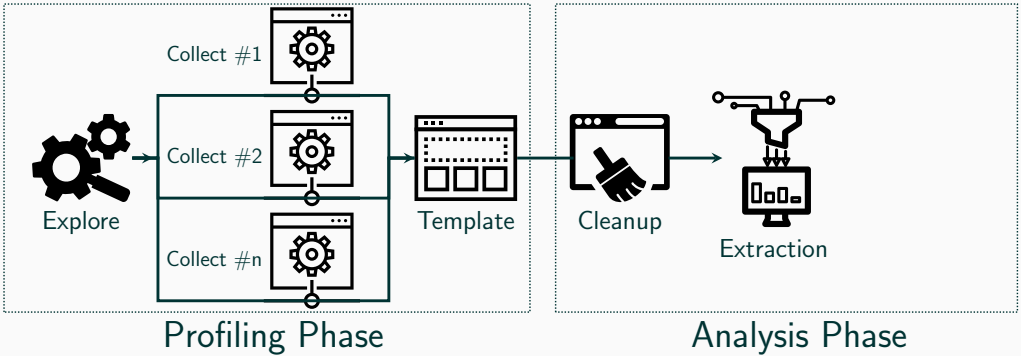
Profiling Phase


Collect #1
Collect #2
Collect #n
Explore
Template
Profiling Phase

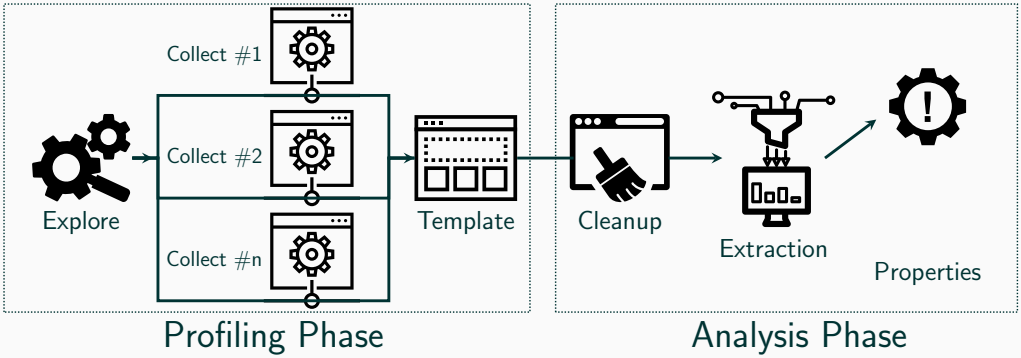
JavaScript Template Attacks - Overview



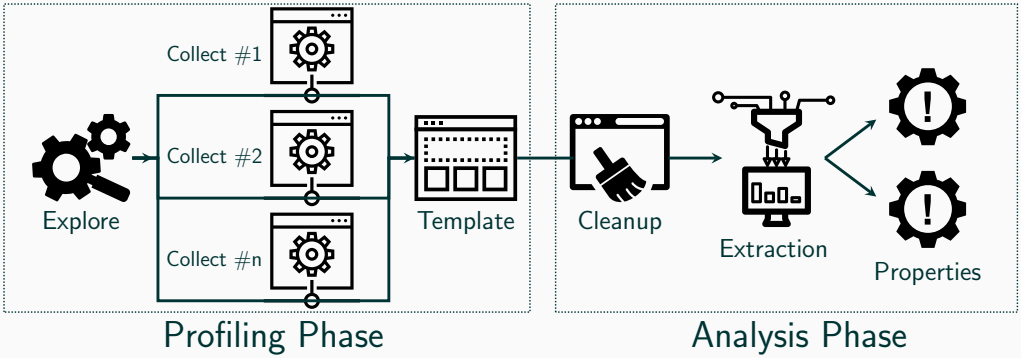
JavaScript Template Attacks - Overview



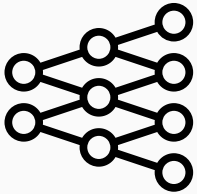
JavaScript Template Attacks - Overview



JavaScript Template Attacks - Overview

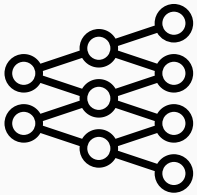


Profiling Phase



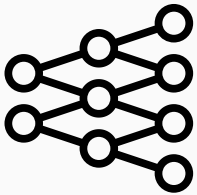
- Exploration of properties

Profiling Phase



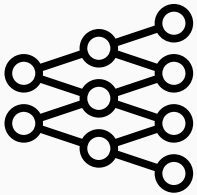
- Exploration of properties
  - Reflections to iterate over all objects

Profiling Phase



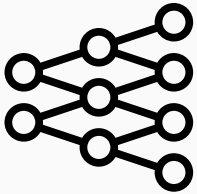
- Exploration of properties
  - Reflections to iterate over all objects
  - Recursively, until all objects are discovered

Profiling Phase



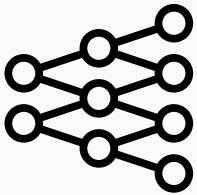
- Exploration of properties
  - Reflections to iterate over all objects
  - Recursively, until all objects are discovered
- Collection of property values

Profiling Phase



- Exploration of properties
  - Reflections to iterate over all objects
  - Recursively, until all objects are discovered
- Collection of property values
  - For every discovered property, acquire value

Profiling Phase



- Exploration of properties
  - Reflections to iterate over all objects
  - Recursively, until all objects are discovered
- Collection of property values
  - For every discovered property, acquire value
  - Repeat with changing environments

Template

- Template is a table, rows are properties, columns are environments

- Template is a table, rows are properties, columns are environments

Property
window.Array.name
window.window.Array.name
navigator.platform
performance.timeOrigin
window.SharedWorker
...

- Template is a table, rows are properties, columns are environments

Property	Environment 1
window.Array.name	Array
window.window.Array.name	Array
navigator.platform	Linux x86_64
performance.timeOrigin	1551003902225
window.SharedWorker	function SharedWorker()
...	...

- Template is a table, rows are properties, columns are environments

Property	Environment 1	Environment 2
window.Array.name	Array	Array
window.window.Array.name	Array	Array
navigator.platform	Linux x86_64	Linux armv7l
performance.timeOrigin	1551003902225	1551003815955
window.SharedWorker	function SharedWorker()	null
...	...	...

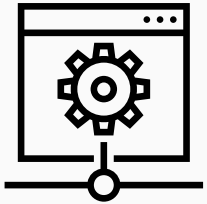
- Template is a table, rows are properties, columns are environments

Property	Environment 1	Environment 2	Environment 3
window.Array.name	Array	Array	Array
window.window.Array.name	Array	Array	Array
navigator.platform	Linux x86_64	Linux armv7l	Win32
performance.timeOrigin	1551003902225	1551003815955	1551003721632
window.SharedWorker	function SharedWorker()	null	function SharedWorker()
...	...	...	...

- Cleanup template (remove duplicates and non-static values)

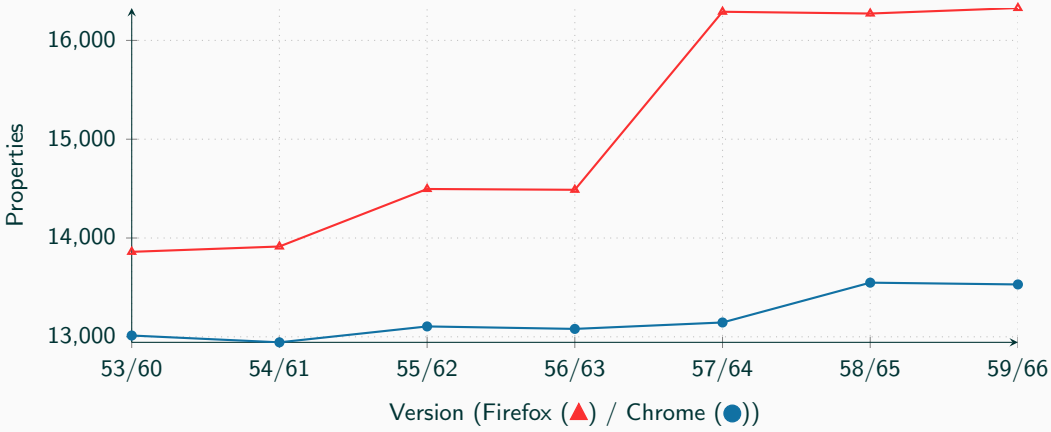
- Cleanup template (remove duplicates and non-static values)

Property	Environment 1	Environment 2	Environment 3
window.Array.name	Array	Array	Array
~~window.window.Array.name~~	~~Array~~	~~Array~~	~~Array~~
navigator.platform	Linux x86_64	Linux armv7l	Win32
~~performance.timeOrigin~~	~~1551003902225~~	~~1551003815955~~	~~1551003721632~~
window.SharedWorker	function SharedWorker()	null	function SharedWorker()
...	...	...	...



Browser	MDN	JavaScript Template
Firefox	2247	15 709
Chrome	2698	13 570
Edge	1806	9666
Firefox Android	2104	15 612
Chrome Android	2676	13 119
Tor browser	2247†	15 639

Properties - Trend

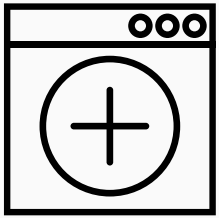


Analysis Phase

- Analyse template (remove values which are the same for all environments)

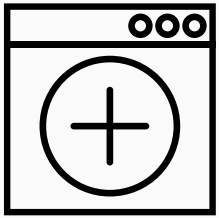
- Analyse template (remove values which are the same for all environments)

Property	Environment 1	Environment 2	Environment 3
~~window.Array.name~~	~~Array~~	~~Array~~	~~Array~~
~~window.window.Array.name~~	~~Array~~	~~Array~~	~~Array~~
navigator.platform	Linux x86_64	Linux armv7l	Win32
~~performance.timeOrigin~~	~~1551003902225~~	~~1551003815955~~	~~1551003721632~~
window.SharedWorker	function SharedWorker()	null	function SharedWorker()
...	...	...	...



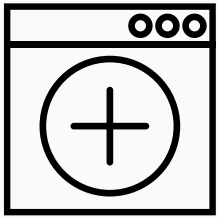
- JavaScript allows defining properties at runtime

Artifical Properties



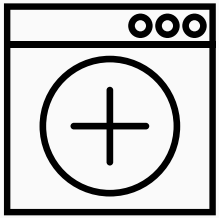
- JavaScript allows defining properties at runtime
- Add “artifical” properties before the profiling phase

Artifical Properties



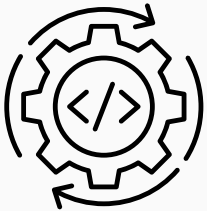
- JavaScript allows defining properties at runtime
- Add “artifical” properties before the profiling phase
- Artificial properties are properties containing results of functions

Artifical Properties



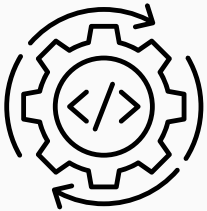
- JavaScript allows defining properties at runtime
- Add “artifical” properties before the profiling phase
- Artificial properties are properties containing results of functions
- → Gather even more information about the environment

JIT Artificial Properties



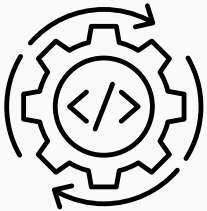
- We show 2 new side channels against the JIT compiler

JIT Artificial Properties



- We show 2 new side channels against the JIT compiler
- Detect internal memory allocator block size

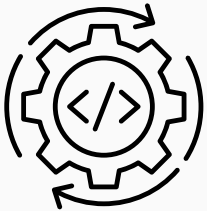
JIT Artificial Properties



- We show 2 new side channels against the JIT compiler
- Detect internal memory allocator block size

→ Timing differences when re-allocating memory

JIT Artificial Properties

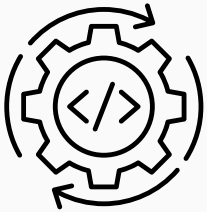


- We show 2 new side channels against the JIT compiler
- Detect internal memory allocator block size

→ Timing differences when re-allocating memory

- Distinguish 32 bit from 64 bit systems

JIT Artificial Properties



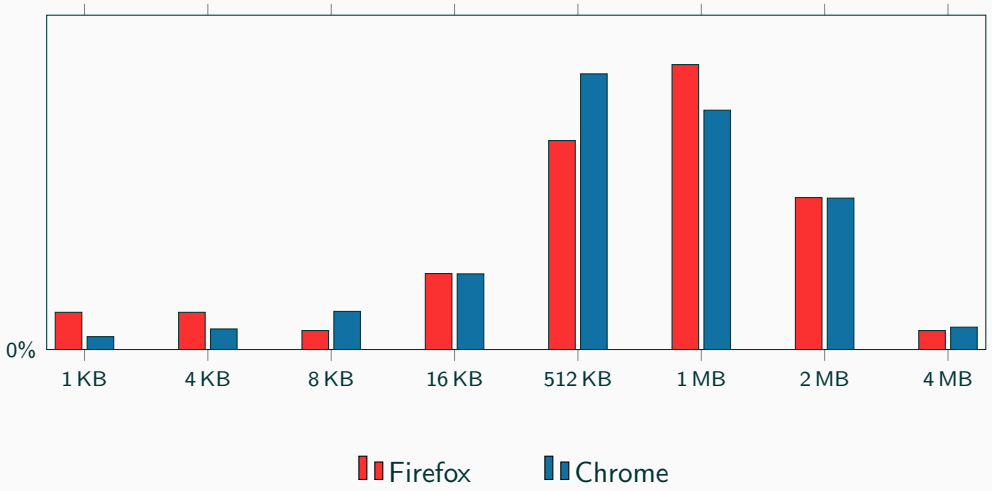
- We show 2 new side channels against the JIT compiler
- Detect internal memory allocator block size

→ Timing differences when re-allocating memory

- Distinguish 32 bit from 64 bit systems

→ JIT can use more registers on 64-bit systems

Memory Allocator Side Channel



```
var a = 0.9, b = c = d = e = f = g
    = 0;
for(var i = 0; i < 10000000; i++)
    {
  b = 1.0 / a;
  c = 2.2 / b;
  d = 3.4 / c;
  e = 4.1 / d;
  f = 5.8 / e;
  g = 6.6 / f;
  // no operation
  a = a + b + c + d + e + f + g +
    g;
}
```

```
var a = 0.9, b = c = d = e = f = g
    = h = 0;
for(var i = 0; i < 10000000; i++)
    {
  b = 1.0 / a;
  c = 2.2 / b;
  d = 3.4 / c;
  e = 4.1 / d;
  f = 5.8 / e;
  g = 6.6 / f;
  h = 7.1 / g;
  a = a + b + c + d + e + f + g +
    h;
}
```

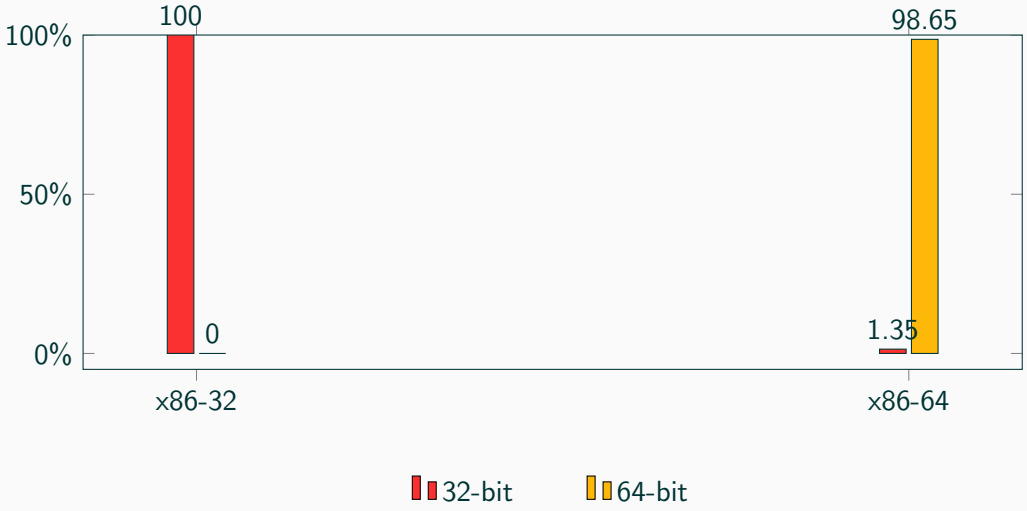
```
vaddss %xmm0,%xmm1,%xmm1
vdivsd %xmm7,%xmm6,%xmm6
vmovsd %xmm7,0x8(%esp)
vxorpd %xmm2,%xmm2,%xmm2
vxorpd %xmm7,%xmm7,%xmm7
```

x86-32

```
vaddsd %xmm0,%xmm1,%xmm0
vdivsd %xmm2,%xmm11,%xmm3
vaddsd %xmm2,%xmm0,%xmm0
vdivsd %xmm3,%xmm10,%xmm4
```

x86-64

ISA Side Channel



Results

- Distinguish browser including exact version

- Distinguish browser including exact version
- Both number and value of properties differ significantly

Browser Detection

- Distinguish browser including exact version
- Both number and value of properties differ significantly
- `toString` as simple artificial property

Browser Detection

- Distinguish browser including exact version
- Both number and value of properties differ significantly
- `toString` as simple artificial property

$\rightarrow$ different string representations

- Distinguish browser including exact version
- Both number and value of properties differ significantly
- `toString` as simple artificial property

→ different string representations

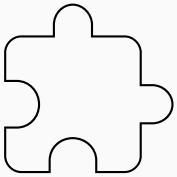
- 5796 different properties between Firefox and Chrome

Browser Detection

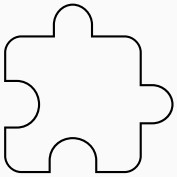
- Distinguish browser including exact version
- Both number and value of properties differ significantly
- `toString` as simple artificial property

→ different string representations

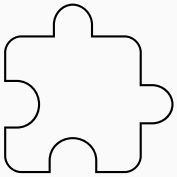
- 5796 different properties between Firefox and Chrome
- Distinguished all 40 tested browsers



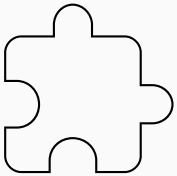
- Most extensions modify or add properties



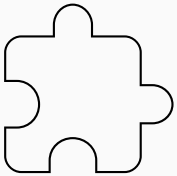
- Most extensions modify or add properties
- Installed privacy extensions (e.g., Canvas Defenser, Ghostery)



- Most extensions modify or add properties
- Installed privacy extensions (e.g., Canvas Defenser, Ghostery)
- Not only presence, but also settings (e.g., protection level)



- Most extensions modify or add properties
- Installed privacy extensions (e.g., Canvas Defenser, Ghostery)
- Not only presence, but also settings (e.g., protection level)
- Canvas Defender only renamed original functions → automatically detected



- Most extensions modify or add properties
- Installed privacy extensions (e.g., Canvas Defenser, Ghostery)
- Not only presence, but also settings (e.g., protection level)
- Canvas Defender only renamed original functions → automatically detected

→ Circumvents extension

More Results

- Private mode, e.g.,

More Results

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)

More Results

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)

More Results

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,

More Results

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)
  - Different font dimensions

More Results

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)
  - Different font dimensions
- CPU vendor (WebGL and ISA side channel)

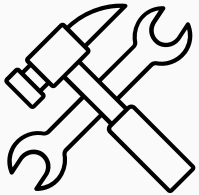
- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)
  - Different font dimensions
- CPU vendor (WebGL and ISA side channel)
- Virtual machine, e.g.,

- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)
  - Different font dimensions
- CPU vendor (WebGL and ISA side channel)
- Virtual machine, e.g.,
  - WebGL vendor (Firefox)

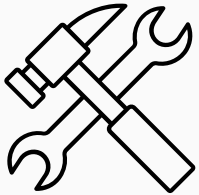
- Private mode, e.g.,
  - Shared workers unavailable (Firefox)
  - Local databases unavailable (Edge)
- Operating system, e.g.,
  - Virtual-reality displays (Windows, partly on macOS)
  - Different font dimensions
- CPU vendor (WebGL and ISA side channel)
- Virtual machine, e.g.,
  - WebGL vendor (Firefox)
  - Strange screen resolution

You can find our proof-of-concept implementation on:

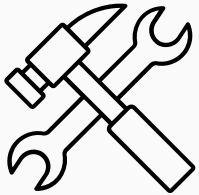
- `https://github.com/IAIK/jstemplate`



- Properties returned by function calls



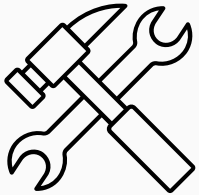
- Properties returned by function calls
- Requires understanding function semantics



- Properties returned by function calls
- Requires understanding function semantics

→ Number and type of arguments, side effects (e.g., close())

Future Work



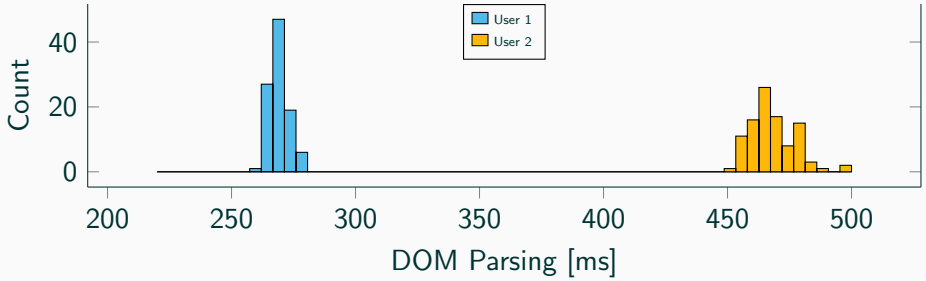
- Properties returned by function calls
- Requires understanding function semantics

→ Number and type of arguments, side effects (e.g., close())

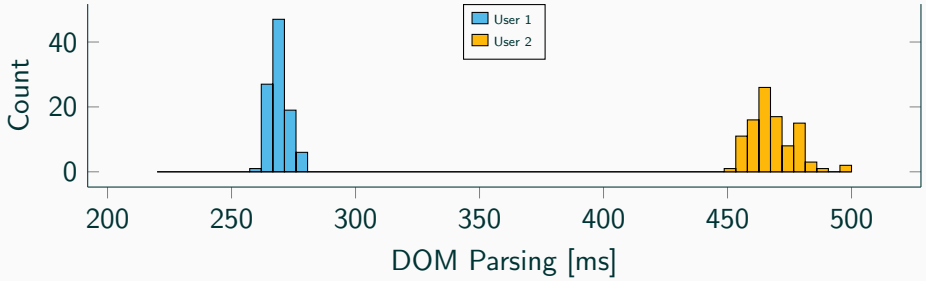
- New web standards (e.g., Web USB, Web NFC)

- Non-static properties can be used as distribution

- Non-static properties can be used as distribution



- Non-static properties can be used as distribution



→ timings depends e.g., on CPU speed

Conclusion

- JavaScript Template attacks detects various environment properties

Conclusion

- JavaScript Template attacks detects various environment properties
- Enables exploits, side-channel attacks and plausible phishing

Conclusion

- JavaScript Template attacks detects various environment properties
- Enables exploits, side-channel attacks and plausible phishing
- Tool for browser vendors to find leakage

- JavaScript Template attacks detects various environment properties
- Enables exploits, side-channel attacks and plausible phishing
- Tool for browser vendors to find leakage
- Advances field of fingerprinting

JavaScript Template Attacks

Michael Schwarz (@misc0110), Florian Lackner, Daniel Gruss (@lavados)

Februrary 25, 2019

IAIK – Graz University of Technology