Memento
Learning Secrets from Process Footprints[3]
Suman Jana  Vitaly Shmatikov

CAP6135 Malware & Software Vulnerability Analysis

Sidhanth Sheelavanth¹
under
Prof. Cliff Zou¹

¹University of Central Florida
Dept. of EECS

April 16, 2014
About Paper

- Authors - Suman Jana, Vitaly Shmatikov. UT - Austin.
- Best Student Paper Award.
- Partly funded by NSF grants.
- Demo - side channel attack.
Outline

1 Introduction
2 Why Do it?
3 Attack Overview.
4 Attack Details.
   • Browser Mem Management.
   • When it works?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   • Advanced Attacks.
   • CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   • Pros.
   • Cons.
10 Appendix.
Outline

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1. Introduction
2. Why Do it ?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works ?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1 Introduction
2 Why Do it ?
3 Attack Overview.
4 Attack Details.
   • Browser Mem Management.
   • When it works ?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   • Advanced Attacks.
   • CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   • Pros.
   • Cons.
10 Appendix.
Outline

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1 Introduction
2 Why Do it?
3 Attack Overview.
4 Attack Details.
   - Browser Mem Management.
   - When it works?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   - Pros.
   - Cons.
10 Appendix.
Outline

1. Introduction
2. Why Do it ?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works ?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Outline

1 Introduction
2 Why Do it ?
3 Attack Overview.
4 Attack Details.
   - Browser Mem Management.
   - When it works ?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   - Pros.
   - Cons.
10 Appendix.
Outline

1. Introduction.
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Introduction

- **Memento**\(^1\) -

  ![Memento poster](image)

**Memento (2000)**

- **Duration:** 113 min
- **Genre:** Mystery, Thriller
- **Release Date:** 11 October 2000 (France)

**Your rating:** ★★★★★★★★★★ 8.6/10

- **Rating:** 8.6/10 from 610,139 users
- **Metascore:** 80/100
- **Reviews:** 1,978 user | 261 critic | 34 from Metacritic.com

A man, suffering from short-term memory loss, uses notes and tattoos to hunt for the man he thinks killed his wife.

**Director:** Christopher Nolan

**Writers:** Christopher Nolan (screenplay), Jonathan Nolan (short story "Memento Mori")

**Stars:** Guy Pearce, Carrie-Anne Moss, Joe Pantoliano | See full cast and crew »
Introduction

Terminology

- Side Channel Attack.
  - [P] Timing(CPU, mem), Power Analysis(SPA, DPA), Acoustic Cryptanalysis, Differential Fault, Data Remanence.[2]

- Secrets - Webpage Identity, Finer grained information.

- Process Footprint - DRS/WS/RSS.

<table>
<thead>
<tr>
<th>Handles</th>
<th>NPM(K)</th>
<th>PM(K)</th>
<th>WS(K)</th>
<th>VM(M)</th>
<th>CPU(s)</th>
<th>Id</th>
<th>ProcessName</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>33</td>
<td>81844</td>
<td>66392</td>
<td>270</td>
<td>4.98</td>
<td>380</td>
<td>chrome</td>
</tr>
<tr>
<td>162</td>
<td>34</td>
<td>62176</td>
<td>49672</td>
<td>241</td>
<td>4.85</td>
<td>3536</td>
<td>chrome</td>
</tr>
<tr>
<td>161</td>
<td>28</td>
<td>52988</td>
<td>40256</td>
<td>216</td>
<td>2.57</td>
<td>4040</td>
<td>chrome</td>
</tr>
<tr>
<td>1610</td>
<td>101</td>
<td>130828</td>
<td>190828</td>
<td>489</td>
<td>2,684.47</td>
<td>6040</td>
<td>chrome</td>
</tr>
<tr>
<td>163</td>
<td>34</td>
<td>57264</td>
<td>45744</td>
<td>230</td>
<td>4.63</td>
<td>6964</td>
<td>chrome</td>
</tr>
<tr>
<td>306</td>
<td>37</td>
<td>183208</td>
<td>185720</td>
<td>505</td>
<td>1,963.55</td>
<td>7268</td>
<td>chrome</td>
</tr>
<tr>
<td>213</td>
<td>41</td>
<td>88940</td>
<td>89224</td>
<td>287</td>
<td>29.33</td>
<td>7396</td>
<td>chrome</td>
</tr>
<tr>
<td>159</td>
<td>24</td>
<td>36772</td>
<td>27980</td>
<td>198</td>
<td>25.29</td>
<td>7444</td>
<td>chrome</td>
</tr>
<tr>
<td>168</td>
<td>28</td>
<td>67800</td>
<td>64464</td>
<td>246</td>
<td>34.12</td>
<td>7480</td>
<td>chrome</td>
</tr>
<tr>
<td>165</td>
<td>35</td>
<td>65516</td>
<td>51644</td>
<td>231</td>
<td>12.26</td>
<td>7600</td>
<td>chrome</td>
</tr>
<tr>
<td>160</td>
<td>25</td>
<td>74088</td>
<td>55730</td>
<td>254</td>
<td>12.50</td>
<td>7746</td>
<td>chrome</td>
</tr>
</tbody>
</table>
Introduction

Terminology

- Side Channel Attack.
  - [P] Timing(CPU, mem), Power Analysis(SPA,DPA), Acoustic Cryptanalysis, Differential Fault, Data Remanence.[2]
- Secrets - Webpage Identity, Finer grained information.
- Process Footprint - DRS/WS/RSS.

```powershell
PS C:\Users\Sid> get-process chrome
```

<table>
<thead>
<tr>
<th>Handles</th>
<th>NPM&lt;K&gt;</th>
<th>PM&lt;K&gt;</th>
<th>WS&lt;K&gt;</th>
<th>VM&lt;M&gt;</th>
<th>CPU&lt;s&gt;</th>
<th>Id</th>
<th>ProcessName</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>33</td>
<td>81844</td>
<td>66392</td>
<td>270</td>
<td>4.98</td>
<td>380</td>
<td>chrome</td>
</tr>
<tr>
<td>162</td>
<td>34</td>
<td>62176</td>
<td>49672</td>
<td>241</td>
<td>4.85</td>
<td>3536</td>
<td>chrome</td>
</tr>
<tr>
<td>161</td>
<td>28</td>
<td>52988</td>
<td>40256</td>
<td>216</td>
<td>2.57</td>
<td>4040</td>
<td>chrome</td>
</tr>
<tr>
<td>1610</td>
<td>101</td>
<td>130828</td>
<td>190828</td>
<td>489</td>
<td>2,684.47</td>
<td>6040</td>
<td>chrome</td>
</tr>
<tr>
<td>163</td>
<td>34</td>
<td>57264</td>
<td>45744</td>
<td>230</td>
<td>4.63</td>
<td>6964</td>
<td>chrome</td>
</tr>
<tr>
<td>306</td>
<td>37</td>
<td>183208</td>
<td>185720</td>
<td>505</td>
<td>1,963.55</td>
<td>7268</td>
<td>chrome</td>
</tr>
<tr>
<td>213</td>
<td>41</td>
<td>88940</td>
<td>89224</td>
<td>287</td>
<td>29.33</td>
<td>7396</td>
<td>chrome</td>
</tr>
<tr>
<td>159</td>
<td>24</td>
<td>36772</td>
<td>27980</td>
<td>198</td>
<td>25.29</td>
<td>7444</td>
<td>chrome</td>
</tr>
<tr>
<td>168</td>
<td>28</td>
<td>67800</td>
<td>64464</td>
<td>246</td>
<td>34.12</td>
<td>7480</td>
<td>chrome</td>
</tr>
<tr>
<td>165</td>
<td>35</td>
<td>65516</td>
<td>51644</td>
<td>231</td>
<td>12.26</td>
<td>7600</td>
<td>chrome</td>
</tr>
<tr>
<td>163</td>
<td>25</td>
<td>74288</td>
<td>55732</td>
<td>254</td>
<td>48.58</td>
<td>7744</td>
<td>chrome</td>
</tr>
</tbody>
</table>
Introduction

Terminology

- Side Channel Attack.
  - [P] Timing(CPU, mem), Power Analysis(SPA,DPA), Acoustic Cryptanalysis, Differential Fault, Data Remanence.[2]

- Secrets - Webpage Identity, Finer grained information.

- Process Footprint - DRS/WS/RSS.

![Process Footprint](image.png)
Side Channel Attack.
- Timing(CPU, mem), Power Analysis(SPA,DPA), Acoustic Cryptanalysis, Differential Fault, Data Remanence.\[^2\]

Secrets - Webpage Identity, Finer grained information.

Process Footprint - DRS/WS/RSS.

```
PS C:\Users\Sid> get-process chrome

<table>
<thead>
<tr>
<th>Handles</th>
<th>NPM(K)</th>
<th>PM(K)</th>
<th>WS(K)</th>
<th>VM(M)</th>
<th>CPU(s)</th>
<th>Id</th>
<th>ProcessName</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>33</td>
<td>81844</td>
<td>66392</td>
<td>270</td>
<td>4.98</td>
<td>380</td>
<td>chrome</td>
</tr>
<tr>
<td>162</td>
<td>34</td>
<td>62176</td>
<td>49672</td>
<td>241</td>
<td>4.85</td>
<td>3536</td>
<td>chrome</td>
</tr>
<tr>
<td>161</td>
<td>28</td>
<td>52988</td>
<td>40256</td>
<td>216</td>
<td>2.57</td>
<td>4040</td>
<td>chrome</td>
</tr>
<tr>
<td>1610</td>
<td>101</td>
<td>130828</td>
<td>190828</td>
<td>489</td>
<td>2.68447</td>
<td>6040</td>
<td>chrome</td>
</tr>
<tr>
<td>163</td>
<td>34</td>
<td>57264</td>
<td>45744</td>
<td>230</td>
<td>4.63</td>
<td>6964</td>
<td>chrome</td>
</tr>
<tr>
<td>306</td>
<td>37</td>
<td>183208</td>
<td>185720</td>
<td>505</td>
<td>1.96355</td>
<td>7268</td>
<td>chrome</td>
</tr>
<tr>
<td>213</td>
<td>41</td>
<td>88940</td>
<td>89224</td>
<td>287</td>
<td>29.33</td>
<td>7396</td>
<td>chrome</td>
</tr>
<tr>
<td>159</td>
<td>24</td>
<td>36772</td>
<td>27980</td>
<td>198</td>
<td>25.29</td>
<td>7444</td>
<td>chrome</td>
</tr>
<tr>
<td>168</td>
<td>28</td>
<td>67800</td>
<td>64464</td>
<td>246</td>
<td>34.12</td>
<td>7480</td>
<td>chrome</td>
</tr>
<tr>
<td>165</td>
<td>35</td>
<td>65516</td>
<td>51644</td>
<td>231</td>
<td>12.26</td>
<td>7600</td>
<td>chrome</td>
</tr>
<tr>
<td>162</td>
<td>25</td>
<td>54228</td>
<td>55738</td>
<td>254</td>
<td>18.50</td>
<td>7744</td>
<td>chrome</td>
</tr>
</tbody>
</table>
```
Outline
YAAAAAWN !!!

1 Introduction.
2 Why Do it ?
3 Attack Overview.
4 Attack Details.
   - Browser Mem Management.
   - When it works ?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   - Pros.
   - Cons.
10 Appendix.
Why Do It?

- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- OS mechanisms increasingly leveraged.
  - Android, Network Daemons, Chrome, IE.
- Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang[^4]. `/proc` ↔ ESP. Keystroke sniffing
  - Dawn Song[^5]. Timing analysis on SSH.
- Different Attack model. *Network Attacker vs Local Attacker.*
Why Do It?

- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- **OS mechanisms increasingly leveraged.**
  - Android, Network Daemons, Chrome, IE.
- **Related Work.** Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang \([4]\) \(\rightarrow\) /proc ↔ ESP. Keystroke sniffing
  - Dawn Song \([5]\) \(\rightarrow\) Timing analysis on SSH.
- **Different Attack model.** *Network Attacker vs Local Attacker.*
Why Do It?

- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- OS mechanisms increasingly leveraged.
  - Android, Network Daemons, Chrome, IE.
- Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang\textsuperscript{[4]} . \texttt{/proc} $\leftrightarrow$ ESP. Keystroke sniffing
  - Dawn Song \textsuperscript{[5]} . Timing analysis on SSH.
- Different Attack model. Network Attacker vs Local Attacker.
Why Do It?

• Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
• OS mechanisms increasingly leveraged.
  • Android, Network Daemons, Chrome, IE.
• Related Work. Fails with non-deterministic programs (ESP not required).
  • Zhang, Wang\textsuperscript{[4]} . /proc $\leftrightarrow$ ESP. Keystroke sniffing
  • Dawn Song \textsuperscript{[5]} . Timing analysis on SSH.
• Different Attack model. Network Attacker vs Local Attacker.
Why Do It?

- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- OS mechanisms increasingly leveraged.
  - Android, Network Daemons, Chrome, IE.
- Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang\(^4\) . /proc ↔ ESP. Keystroke sniffing
  - Dawn Song\(^5\) . Timing analysis on SSH.
- Different Attack model. Network Attacker vs Local Attacker.
Why Do It?

- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- OS mechanisms increasingly leveraged.
  - Android, Network Daemons, Chrome, IE.
- Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang\textsuperscript{[4]} \textit{/proc} ↔ ESP. Keystroke sniffing
  - Dawn Song\textsuperscript{[5]} . Timing analysis on SSH.
- Different Attack model. Network Attacker vs Local Attacker.
Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).

OS mechanisms increasingly leveraged.
  - Android, Network Daemons, Chrome, IE.

Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang, Wang[^4]. `/proc ↔ ESP. Keystroke sniffing
  - Dawn Song[^5]. Timing analysis on SSH.

Different Attack model. *Network* Attacker vs *Local* Attacker.
Outline
MUST.. KEEP .. EYes .. op..zzz..

1. Introduction
2. Why Do it ?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works ?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Attack Overview

- 2 Processes in parallel on same host as different users.

1. **Run concurrently**
2. Measure target’s memory footprint (memprint) periodically.
3. Build Signature Database D
4. Perform Attack

- **Attack Process**
  - (Unprivileged, no root access)
  - Ex. *game/utility*
  - *workstation*

- **Victim Process**
  - (Browsing)
  - Ex. *Android Browser*
  - *malicious user*
Attack Overview

- 2 Processes in parallel on same host as different users.

1. Run concurrently.
2. Measure target’s memory footprint (memprint) periodically.
3. Build Signature Database D.
4. Perform Attack.
2 Processes in parallel on same host as different users.

1. Run concurrently.
2. Measure target’s memory footprint (memprint) periodically.
3. Build Signature Database D.
4. Perform Attack.
2 Processes in parallel on same host as different users.

1. Run concurrently.
2. Measure target’s memory footprint (memprint) periodically.
3. Build Signature Database D
4. Perform Attack
2 Processes in parallel on same host as different users.

1. **Run concurrently**.
2. **Measure target’s memory footprint** (memprint) periodically.
3. **Build Signature Database D**.
4. **Perform Attack**.
Attack Overview

2 Processes in parallel on same host as different users.

1. Run concurrently.
2. Measure target’s memory footprint (memprint) periodically.
3. Build Signature Database D.
4. Perform Attack.
Outline
life, the universe and everything ? 42 .

1 Introduction
2 Why Do it ?
3 Attack Overview.
4 Attack Details.
   • Browser Mem Management.
   • When it works ?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   • Advanced Attacks.
   • CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   • Pros.
   • Cons.
10 Appendix.
Attack Details
Browser Mem Management

- Different browsers, different allocators (jemalloc, tcmalloc, etc...).

- Allocator optimization & behaviour, Sensitivity.
- Not directly translated, Varies, Memprint, Noise.
Attack Details
When it works?

- Diversity.
- Stability.
- Which process to monitor?
  - Monolithic browsers.
  - Micro Kernel browsers.
- Network attacks.
Outline

progress bar at the top says 50%. YES!!

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Experimental Setup

- **Browsers** - Chrome, Firefox, Android
- **OS** - Windows, Linux, Android.
- Memory Signature gathering by automated scripts.
- ALEXA top 100,000 websites.
- **Memprint** statistics collected.
  - DRS change recorded using PID.
  - Scaled to 100,000 webpages, attacker pauses victim.
  - *FixSched, Attack*.
- Plugins, addons, extensions alter in predictable ways. Offset calculated or blocker used.
Experimental Setup

- Browsers - Chrome, Firefox, Android
- OS - Windows, Linux, Android.
- Memory Signature gathering by automated scripts.
- ALEXA top 100,000 websites.
- Memprint statistics collected.
  - DRS change recorded using PID.
  - Scaled to 100,000 webpages, attacker pauses victim.
  - FixSched, Attack.
- Plugins, addons, extensions alter in predictable ways. Offset calculated or blocker used.
Experimental Setup

- Browsers - Chrome, Firefox, Android
- OS - Windows, Linux, Android.
- Memory Signature gathering by automated scripts.
- ALEXA top 100,000 websites.
- **Memprint** statistics collected.
  - DRS change recorded using PID.
  - Scaled to 100,000 webpages, attacker pauses victim.
  - *FixSched, Attack.*
- Plugins, addons, extensions alter in predictable ways. Offset calculated or blocker used.
Experimental Setup

- Browsers - Chrome, Firefox, Android
- OS - Windows, Linux, Android.
- Memory Signature gathering by automated scripts.
- ALEXA top 100,000 websites.
- Memprint statistics collected.
  - DRS change recorded using PID.
  - Scaled to 100,000 webpages, attacker pauses victim.
  - FixSched, Attack.
- Plugins, addons, extensions alter in predictable ways. Offset calculated or blocker used.
Experimental Setup

- Browsers - Chrome, Firefox, Android
- OS - Windows, Linux, Android.
- Memory Signature gathering by automated scripts.
- ALEXA top 100,000 websites.
- **Memprint** statistics collected.
  - DRS change recorded using PID.
  - Scaled to 100,000 webpages, attacker pauses victim.
  - *FixSched, Attack.*
- Plugins, addons, extensions alter in predictable ways. Offset calculated or blocker used.
Outline
maybe if I start clapping early .. hel stop ?

1 Introduction
2 Why Do it ?
3 Attack Overview.
4 Attack Details.
   - Browser Mem Management.
   - When it works ?
5 Experimental Setup.
6 Results.
7 Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8 Defenses.
9 Presenter’s Notes.
   - Pros.
   - Cons.
10 Appendix.
Results
Verification

- False +, False -.
- Distinguishability.
  - wrt fixed ambiguity sets.
  - distinguishability = \( (\mu - \sigma) - (\mu_{false} + \sigma_{false}) \)
  - positive or negative?
- Recognizability
  - true positive rate. Not every page produces a match.
  - Fixsched and Attack visited 5-15 times.
  - Threshold = highest
    \( J(sig_p, memprint(visit{to}ambiguity{page})) \).
- Factors affecting accuracy of measurement. (method, concurrent workload, measurement rate, variations).
Results
Verification

- False + , False -.
- Distinguishability  
  - wrt fixed ambiguity sets.
  - distinguishability $= (\mu - \sigma) - (\mu_{false} + \sigma_{false})$
  - positive or negative ?  

- Recognizability  
  - true positive rate. Not every page produces a match.
  - Fixsched and Attack visited 5-15 times.
  - Threshold = highest 
    $J(sig_p, memprint(visittoambiguitypage))$. 

- Factors affecting accuracy of measurement. (method, concurrent workload, measurement rate, variations).
Results
Verification

- False + , False -.
- Distinguishability.
  - wrt fixed ambiguity sets.
  - distinguishability = \((\mu - \sigma) - (\mu_{false} + \sigma_{false})\)
  - positive or negative?

- Recognizability
  - true positive rate. Not every page produces a match.
  - Fixsched and Attack visited 5-15 times.
  - Threshold = highest
    \(J(sig_p, memprint(visittoambiguitypage))\).

- Factors affecting accuracy of measurement. (method, concurrent workload, measurement rate, variations).
Results
Verification

- False +, False -.
- Distinguishability.
  - wrt fixed ambiguity sets.
  - distinguishability = \((\mu - \sigma) - (\mu_{false} + \sigma_{false})\)
  - positive or negative?   

- Recognizability
  - true positive rate. Not every page produces a match.
  - Fixsched and Attack visited 5-15 times.
  - Threshold = highest
    \(J(sig_p, memprint(visittoambiguitypage))\).

- Factors affecting accuracy of measurement. (method, concurrent workload, measurement rate, variations).
Outline

wonder if I have anterograde amnesia ?...

1. Introduction
2. Why Do it ?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works ?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Extensions of Attack
Advanced Attacks

- Variations.
- Web Sessions.
- Similar memprint disambiguation.

[3] Fig. 19, 23
Extensions of Attack
CPU Scheduling Stats

- ESP, keystroke timing relation.\(^4\)
- Use this to differentiate.

\(^3\)Fig. 5, Table V

![INTER-KEYSTROKE TIMINGS IN MILISECONDS: KEYLOGGER VS. STATUS MEASUREMENTS (ANDROID).](image)

![Figure 25. Context-switch delays (LIME in Android).](image)
Outline

hmm. 8’s my new fav number.

1. Introduction
2. Why Do it?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Defenses

- **Changing the OS.**
  - Not a OS specific attack.
  - Can be calculated.
  - Designers must cooperate.

- **Changing the application.**
  - Browser defenses (network, proxy, incognito etc ...) don't work.
  - Reduce app↔OS correlation.
  - Kernel hardening patches.
  - Memory usage abstraction.
  - Monolithic browsers.
Defenses

- Changing the OS.
  - Not a OS specific attack.
  - Can be calculated.
  - Designers must cooperate.

- Changing the application.
  - Browser defenses (network, proxy, incognito etc ...) dont work.
  - Reduce app↔OS correlation.
  - Kernel hardening patches.
  - Memory usage abstraction.
  - monolithic browsers.
Outline

1. Introduction
2. Why Do it ?
3. Attack Overview.
4. Attack Details.
   - Browser Mem Management.
   - When it works ?
5. Experimental Setup.
6. Results.
7. Extensions of Attack.
   - Advanced Attacks.
   - CPU Scheduling Stats.
8. Defenses.
9. Presenter’s Notes.
   - Pros.
   - Cons.
10. Appendix.
Presenters Notes

Pros

- Novel side-channel attack. (Elaborate, complete).
- Proved Hypothesis.
- Structured, well written and precise.
Elaborate attack, result is identity.

- Complexity.
  - Space - $\mathcal{O}(nmw)$.
  - Time - $\mathcal{O}(n^2)$.

- Solutions not concrete.
  - Asynchronous CPUs.
  - Blinding.

- Combination with other side-channel attacks.
  - Network attacks don’t work.
• Elaborate attack, result is identity.
• Complexity.
  • Space - $\mathcal{O}(nmw)$.
  • Time - $\mathcal{O}(n^2)$.
• Solutions not concrete.
  • Asynchronous CPUs.
  • blinding.
• Combination with other side-channel attacks.
  • Network attacks don't work.
Elaborate attack, result is identity.

Complexity.
- Space - $O(nmw)$.
- Time - $O(n^2)$.

Solutions not concrete.
- Asynchronous CPUs.
- Blinding.

Combination with other side-channel attacks.
- Network attacks don’t work.
Elaborate attack, result is identity.

Complexity.
- Space - $\mathcal{O}(nmw)$.
- Time - $\mathcal{O}(n^2)$.

Solutions not concrete.
- Asynchronous CPUs.
- blinding.

Combination with other side-channel attacks.
- Network attacks don’t work.
Elaborate attack, result is identity.

Complexity.
  - Space - $\mathcal{O}(nmw)$.
  - Time - $\mathcal{O}(n^2)$.

Solutions not concrete.
  - Asynchronous CPUs.
  - Blinding.

Combination with other side-channel attacks.
  - Network attacks don’t work.
References

Don’t forget to watch.

QUESTIONS?
Size of Target’s Mem Footprint

- Only info needed is mem size.
- Most OS’s have no restriction on this.
  - Different OS
    - Windows
      - PDH Library
      - cmdlets, get-process(wss,host stats)
    - Linux
      - DRS field in /proc/<pid>/statm
      - Data(mmap) + heap(brk) + code(stack)
      - mm→total_vm - shared_vm
    - Android
      - ps,manifest,kvm_getprocs
Size of Target’s Mem Footprint

- Only info needed is mem size.
- Most OS’s have no restriction on this.

- **Different OS**
  - **Windows**
    - PDH Library
    - `cmdlets, get-process(wss,host stats)`
  - **Linux**
    - DRS field in `/proc/<pid>/statm`
    - `Data(mmap) + heap(brk) + code(stack)`
    - `mm → total_vm - shared_vm`
  - **Android**
    - `ps,manifest,kvm_getprocs`
Size of Target’s Mem Footprint

- Only info needed is mem size.
- Most OS’s have no restriction on this.
  - Different OS
    - Windows
      - PDH Library
        - cmdlets, get-process(wss, host stats)
    - Linux
      - DRS field in /proc/<pid>/statm
      - Data(mmap) + heap(brk) + code(stack)
        - mm→total_vm - shared_vm
    - Android
      - ps, manifest, kvm_getprocs
**Building Signature Database**

- Create *Attack Signatures*, Build Database.
  - Visit $w$ pages $n$ times.
  - Calculate $\text{memprint} = (E, e)$,
  - $E =$int footprint size. (DRS, 6th field of proc), $e =$frequency.
- Comparison of memprints.
  - $((E, e_1) \in \mathcal{m}_1) \land ((E, e_2) \in \mathcal{m}_2) \implies (E, \min(e_1, e_2)) \in \mathcal{m}_1 \cap \mathcal{m}_2$
  - $((E, e_1) \in \mathcal{m}_1) \land ((E, e_2) \in \mathcal{m}_2) \implies (E, \max(e_1, e_2)) \in \mathcal{m}_1 \cup \mathcal{m}_2$
- Similarity using jaccard index.
  $$J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$$
Building Signature Database

- Create *Attack Signatures*, Build Database.
  - Visit $w$ pages $n$ times.
  - Calculate *memprint* $= (E, e)$,
    $E =$int footprint size.(DRS,6th field of proc), $e =$frequency.

- Comparison of memprints.
  - $((E, e_1) \epsilon m_1) \land ((E, e_2) \epsilon m_2) \Rightarrow (E, \min(e_1, e_2)) \epsilon m_1 \cap m_2$
  - $((E, e_1) \epsilon m_1) \land ((E, e_2) \epsilon m_2) \Rightarrow (E, \max(e_1, e_2)) \epsilon m_1 \cup m_2$

- Similarity using jaccard index.
  $J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$

*Overview*
Building Signature Database

- Create *Attack Signatures*, Build Database.
  - Visit $w$ pages $n$ times.
  - Calculate **memprint** = $(E, e)$,
  - $E =$int footprint size.(DRS,6th field of proc), $e =$frequency.

- Comparison of memprints.
  - $((E, e_1) \epsilon m_1) \wedge ((E, e_2) \epsilon m_2) \implies (E, min(e_1, e_2)) \epsilon m_1 \cap m_2$
  - $((E, e_1) \epsilon m_1) \wedge ((E, e_2) \epsilon m_2) \implies (E, max(e_1, e_2)) \epsilon m_1 \cup m_2$

- Similarity using jaccard index.
  - $J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$
Perform Attack

Attack memprint is matched against signature database.

Algorithm 1 Main steps of the matching algorithm

Input: Signature database $D$, attack memprint $s_m$
Output: Matched page or no match

for each page $p$ in $D$ do
  for each signature $sig_p$ for page $p$ in $D$ do
    if $J(s_m, sig_p) > threshold$ then
      Return matched page $p$
    end if
  end for
end for

Return no match
Allocators

valgrind
--smc-check=all --trace-children=yes
--tool=massif
--pages-as-heap=yes --detailed-freq=1
--threshold=0.5
--alloc-fn=mmap
--alloc-fn=syscall
--alloc-fn=pages_map
--alloc-fn=chunk_alloc
--alloc-fn=arena_run_alloc
--alloc-fn=arena_bin_malloc_hard
--alloc-fn=malloc
--alloc-fn=realloc
--alloc-fn='operator new(unsigned long)
--alloc-fn=malloc
--alloc-fn=posix_memalign
--alloc-fn=malloc
--alloc-fn=JS_ArenaAllocate
--alloc-fn=PL_ArenaAllocate
--alloc-fn=NS_Aloc_P
--alloc-fn=NS_Realloc_P
--alloc-fn='XPConnectGCChunkAllocate'
--alloc-fn='PickChunk(JSRuntime*)'
--alloc-fn='RefillFinalizableFreeList'
--alloc-fn=sqite3MemMalloc
--alloc-fn=mallocWithAlarm
--alloc-fn=sqite3Malloc

Figure 2. Firefox: Distribution of malloc'd block sizes.
Distinguishability

Figure 7. Chrome: Distinguishability of 1,000 random pages, 100,000-page ambiguity set (FixSched measurement). 43% of sites are distinguishable.

Figure 11. Chrome and Firefox: Average recognition rate vs. false positive rate for 1,000 pages, 10 visits each, with a 20,000-page (Chrome) and 10,000-page (Firefox) ambiguity set (FixSched measurement).
Figure 14. Chrome and Firefox: Recognizability of 100 random distinguishable pages (Attack and FixSched measurements). No false positives.