Memento

Learning Secrets from Process Footprints[3]

Suman Jana Vitaly Shmatikov

CAP6135 Malware & Software Vulnerability Analysis

Sidhanth Sheelavanth[1]
under
Prof. Cliff Zou[1]

[1]University of Central Florida
Dept. of EECS

April 16, 2014

About Paper

- Authors - Suman Jana , Vitaly Shmatikov. UT - Austin.
- IEEE Symposium on Security and Privacy 2012.
- Best Student Paper Award.
- Partly funded by NSF grants.
- Demo - side channel attack.

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Introduction

- Memento[1] -

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.
- [P]

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

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

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.
- [P]

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

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

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.
- [P]

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

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

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.
- [P]

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

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

Outline
YAAAAWN !!!

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] . */proc* $\leftrightarrow$ 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] . */proc* $\leftrightarrow$ 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] . */proc* $\leftrightarrow$ 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] . */proc* $\leftrightarrow$ 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] . */proc* $\leftrightarrow$ ESP. Keystroke sniffing
  - Dawn Song [5] . Timing analysis on SSH.
- Different Attack model. *Network* Attacker vs *Local* Attacker.

Why Do It ?

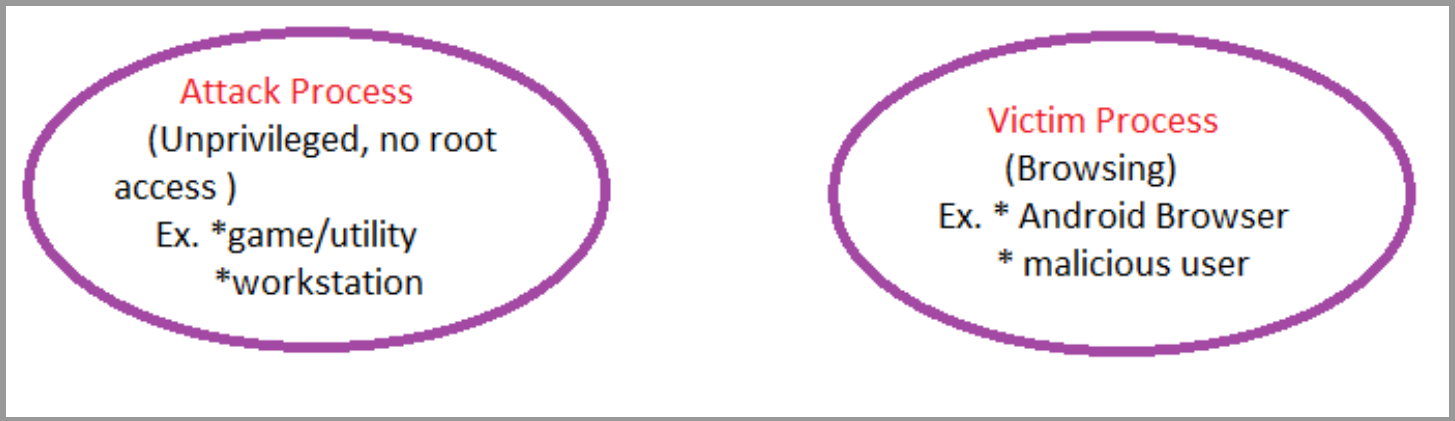
- Symptom of larger problem. Illusion of harmlessness (System isolation mechanisms).
- OS mechanisms increasingly leveraged.
  - Android, Network Daemons,Chrome, IE.
- Related Work. Fails with non-deterministic programs (ESP not required).
  - Zhang,Wang[4] . */proc* $\leftrightarrow$ ESP. Keystroke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

Outline

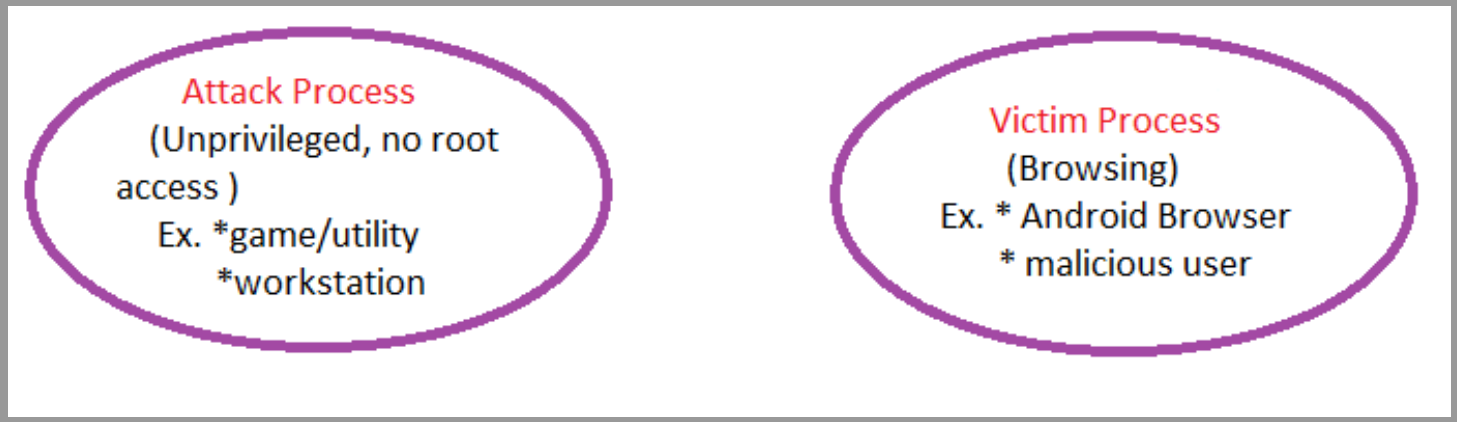
MUST.. KEEP .. EYes .. op..zzz..

Attack Overview

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



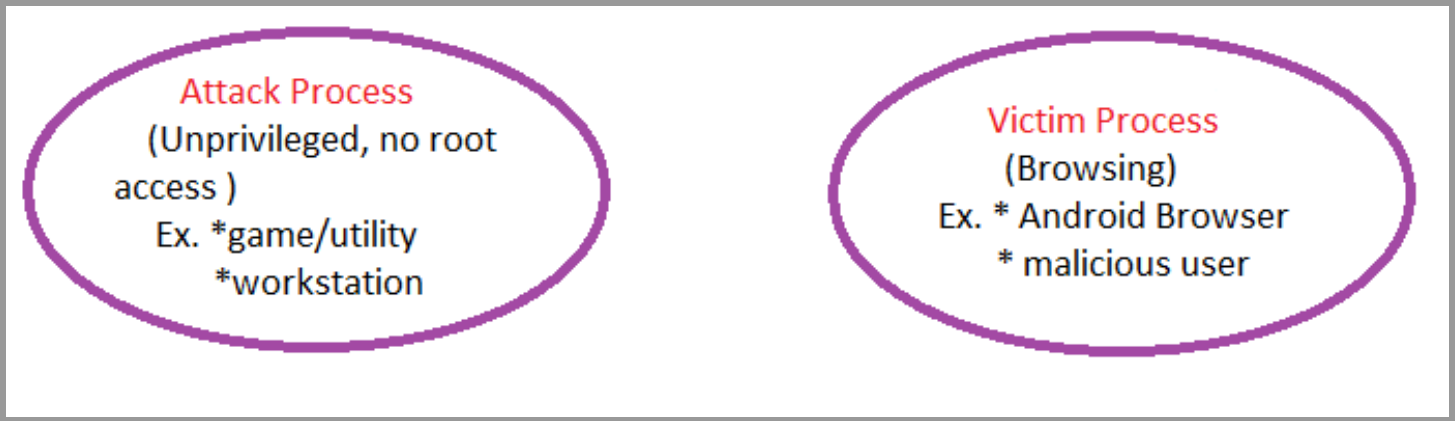
Attack Overview

- 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 Overview

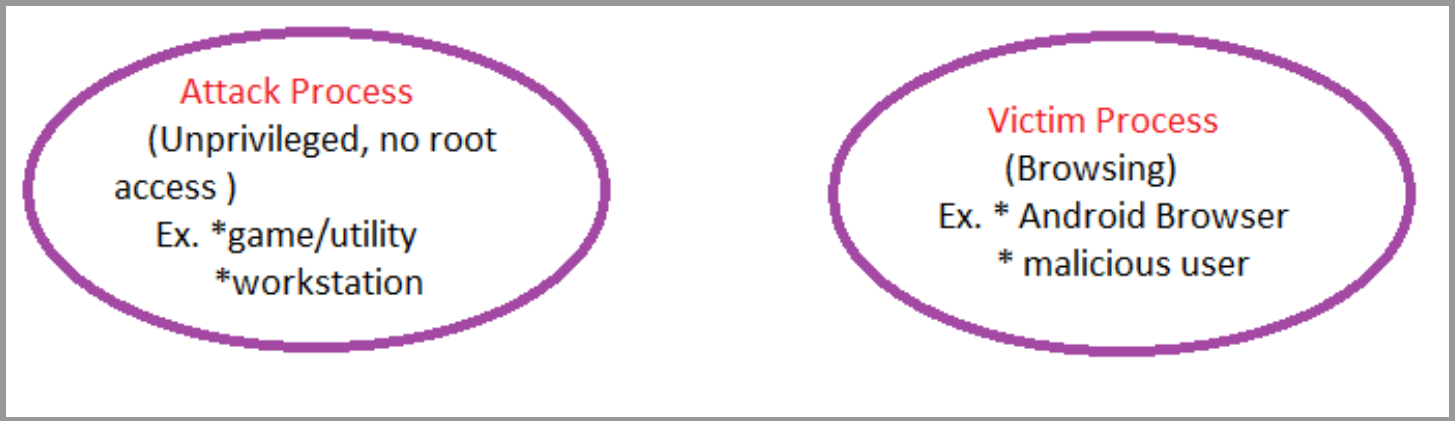
- 2 Processes in parallel on same host as different users.



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

Attack Overview

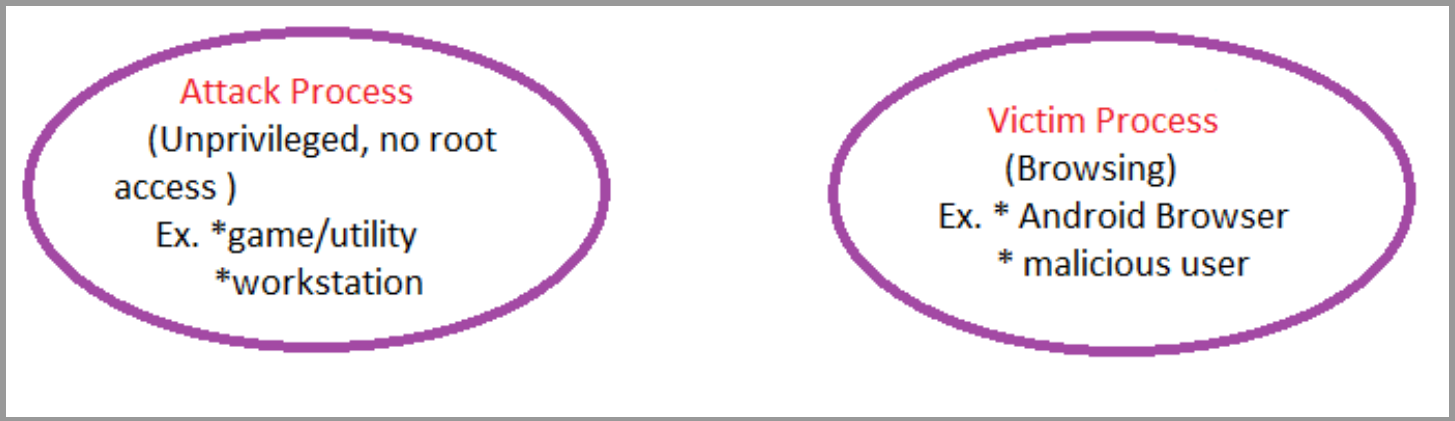
- 2 Processes in parallel on same host as different users.



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

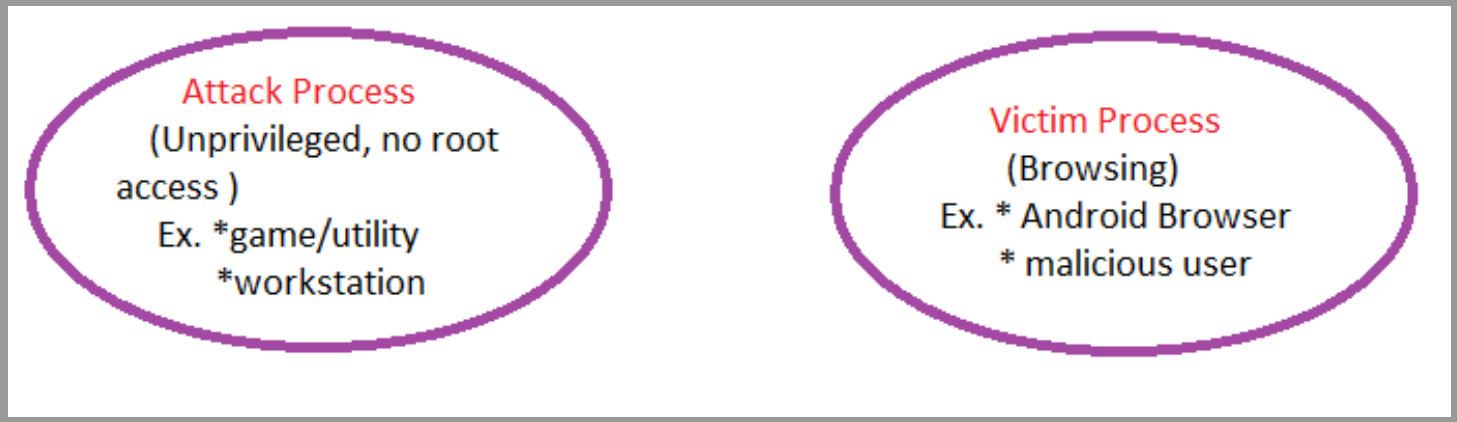
Attack Overview

- 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 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.

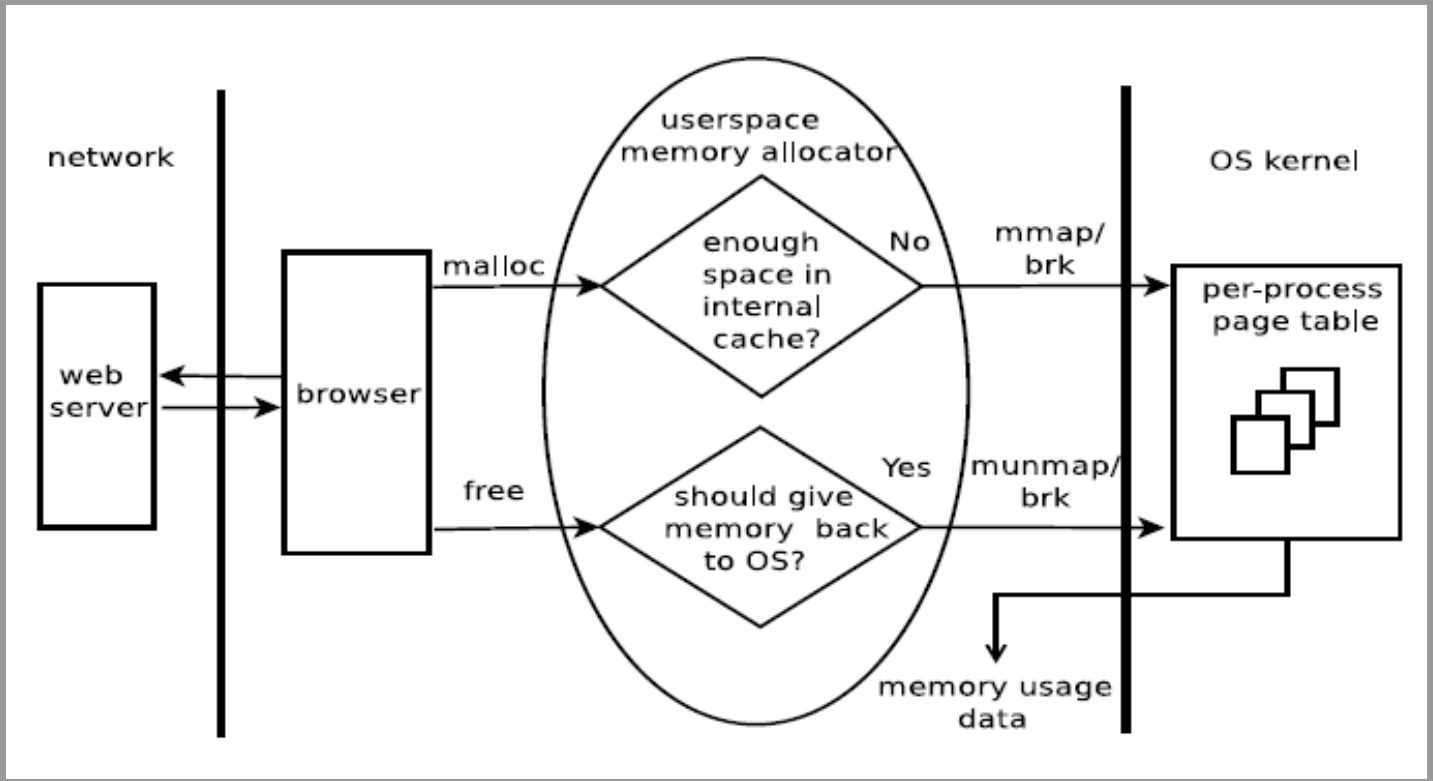
Outline

life, the universe and everything ? 42 .

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!!

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 ?

Results
Verification

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

Outline

wonder if I have anterograde amnesia ?...

Extensions of Attack

Advanced Attacks

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

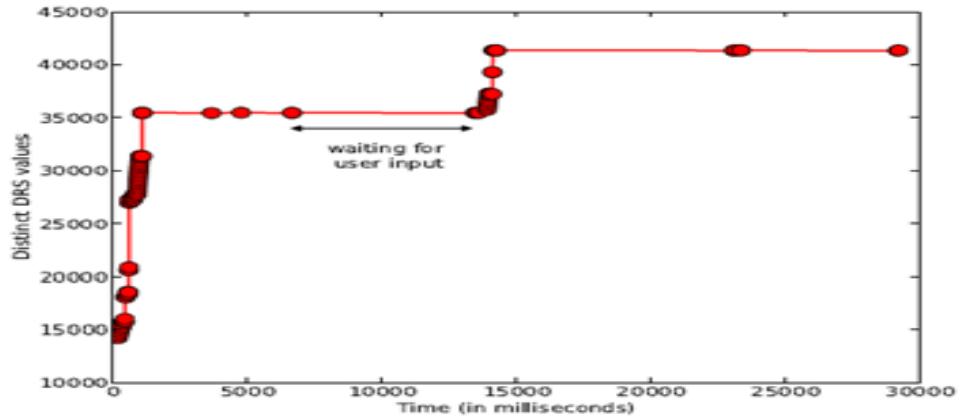


Figure 19. Evolution of the Firefox memory footprint during a Google search session.

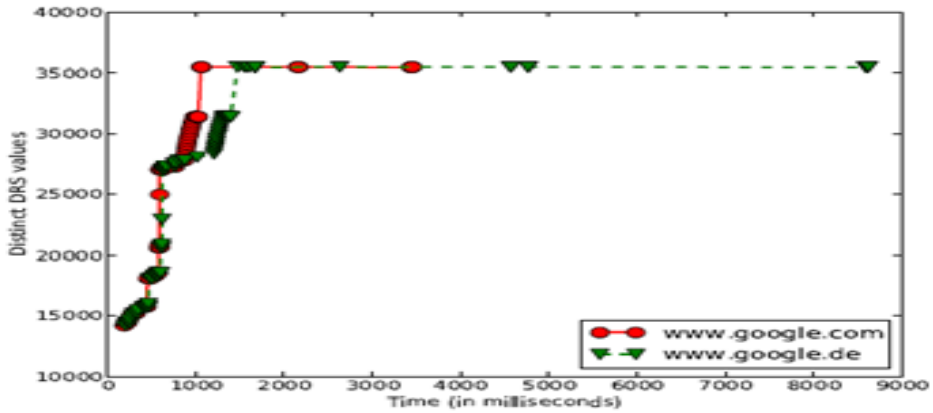


Figure 23. Evolution of the Firefox memory footprint when loading google.com and google.de (US-based browser).

[3] *Fig.*19, 23

Extensions of Attack
CPU Scheduling Stats

- ESP, keystroke timing relation.[4]
- $top - ltm1$,context switches,*schedstat*,Android
- Use this to differentiate.

[3] $Fig.5, TableV$

INTER-KEYSTROKE TIMINGS IN MILLISECONDS: KEYLOGGER VS. STATUS MEASUREMENTS (ANDROID).

Timings	MMS app		bash	
	True	Measured	True	Measured
1	445	449	256	256
2	399	399	320	320
3	176	176	165	175
4	236	240	393	391
5	175	173	255	256

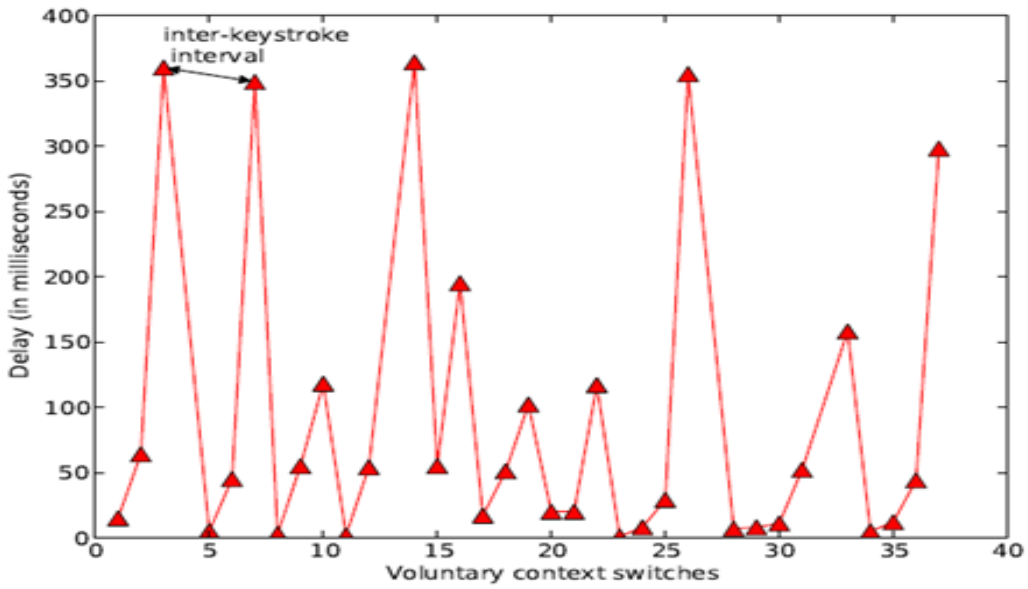


Figure 25. Context-switch delays (LIME in Android).

Outline

hmm. 8 's my new fav number.

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.

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$\leftrightarrow$OS correlation.
  - Kernel hardening patches.
  - Memory usage abstraction.
  - monolithic browsers.

Outline

nope. It's 9...

[P]Presenter's Notes

Pros

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

[P]Presenter's Notes

Cons

- 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.

[P]Presenter's Notes

Cons

- 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.

[P]Presenter's Notes

Cons

- 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.

[P]Presenter's Notes

Cons

- 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.

[P]Presenter's Notes

Cons

- 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

[1] `http://www.imdb.com/title/tt0209144/`

[2] `www.wikipedia.com`

[3] Jana,Suman and Shmatikov,Vitaly, *Memento: Learning Secrets from Process Footprints* in IEEE symposium,2012.

[4] K. Zhang and X. Wang. *Peeping Tom in the neighborhood: Keystroke eavesdropping on multi-user systems.* In USENIX Security, 2009.

[5] D. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In USENIX Security, 2001.

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
```

◂ Overview

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

◂ Overview

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
```

◂ 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.
  $J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$

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.
  $J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$

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.
  $J(m_1, m_2) = \frac{|m_1 \cap m_2|}{|m_1 \cup m_2|}$

Overview

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

◂ Overview

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 lon
--alloc-fn=huge_malloc \
--alloc-fn=posix_memalign \
--alloc-fn=moz_xmalloc \
--alloc-fn=JS_ArenaAllocate \
--alloc-fn=PL_ArenaAllocate \
--alloc-fn=NS_Alloc_P \
--alloc-fn=NS_Realloc_P \
--alloc-fn='XPConnectGCChunkAllocator
--alloc-fn='PickChunk(JSRuntime*)' \
--alloc-fn='RefillFinalizableFreeList
--alloc-fn=sqlite3MemMalloc \
--alloc-fn=mallocWithAlarm \
--alloc-fn=sqlite3Malloc \
<insert-firefox-command-here>
```

◂ Attack Details

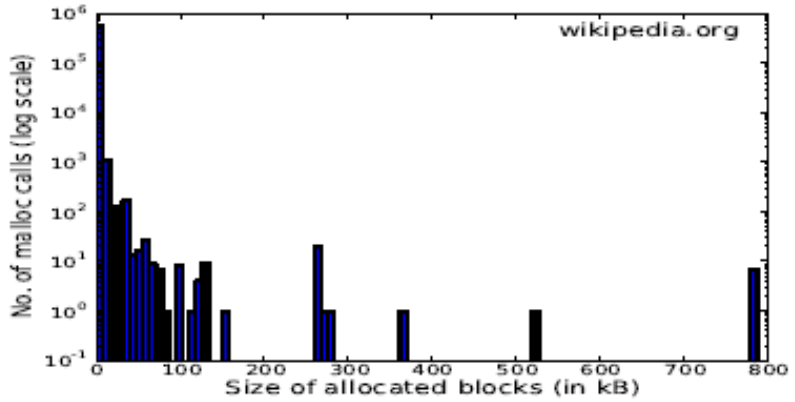
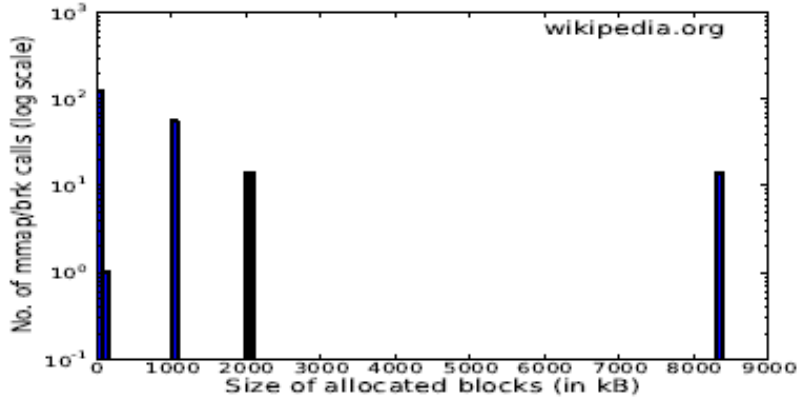


Figure 2. Firefox: Distribution of malloc'd block sizes.



Distinguishability

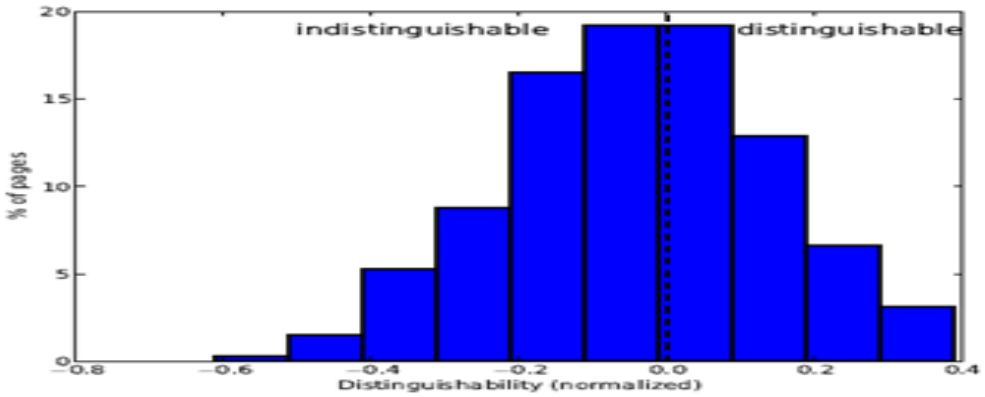


Figure 7. Chrome: Distinguishability of 1,000 random pages, 100,000-page ambiguity set (FixSched measurement). 43% of sites are distinguishable.

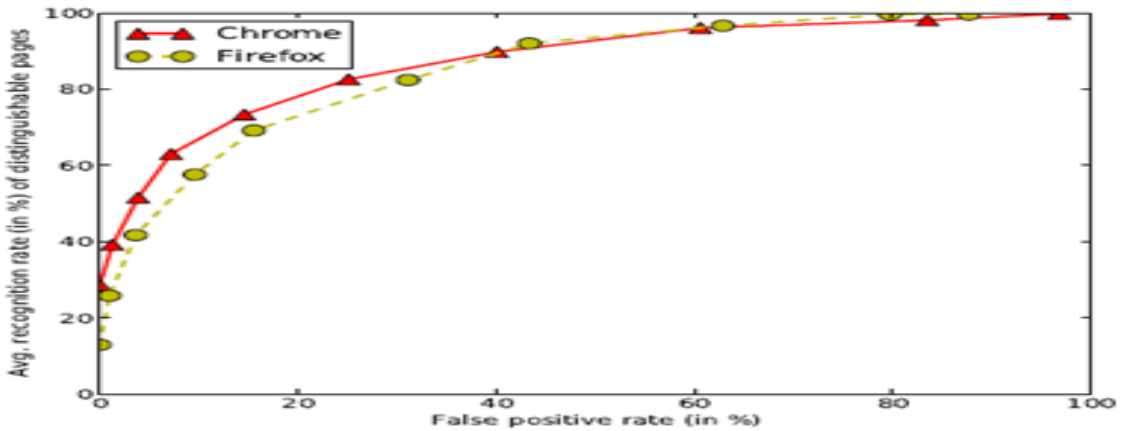


Figure 11. Chrome and Firefox: Average recognition rate vs. false positive rate for 1,000 pages, 10 visits each, with a 20,000-page (Chrome) and 10,000-page (Firefox) ambiguity set (FixSched measurement).

◂ Verification

Recognizability

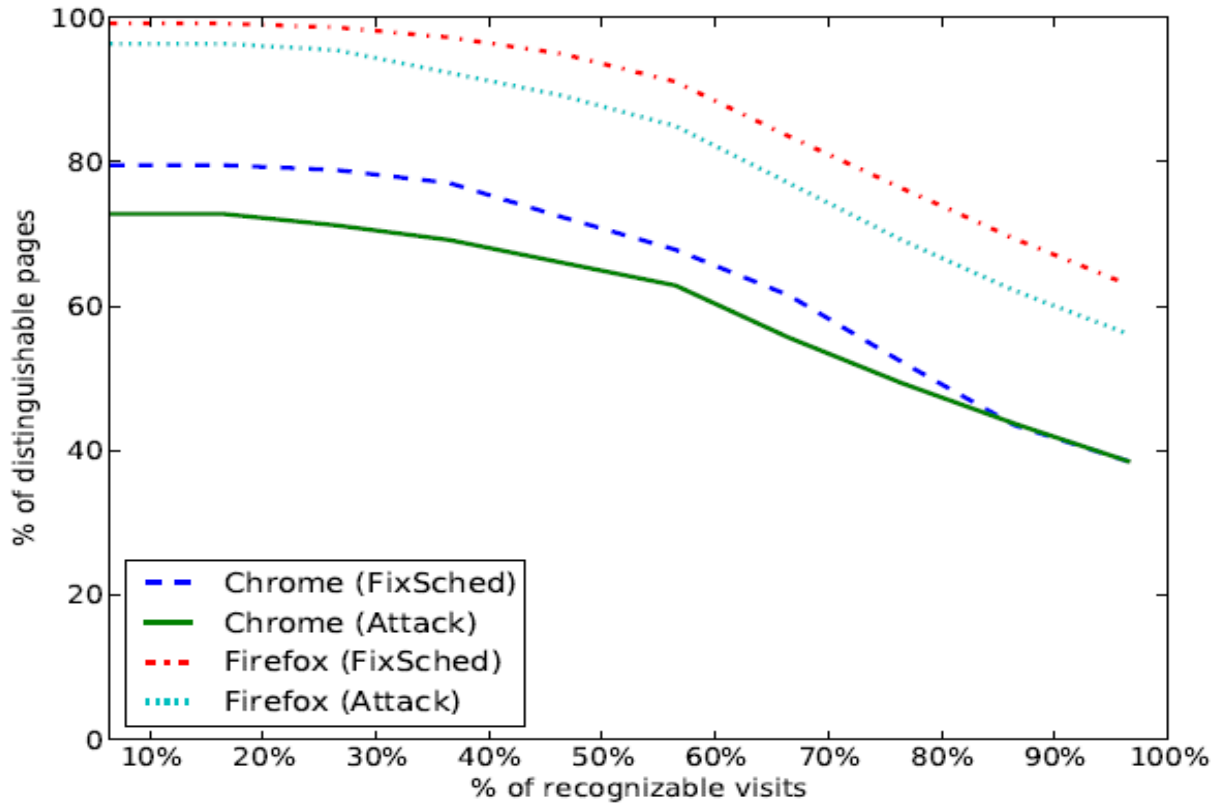


Figure 14. Chrome and Firefox: Recognizability of 100 random distinguishable pages (Attack and FixSched measurements). No false positives.

◂ Verification