What Are You Searching For?
A Remote Keylogging Attack on Search Engine Autocomplete

Vinnie Monaco
Naval Postgraduate School
Search engine autocomplete

Search query

Packet capture
20 years of network side channels

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

Remote timing attacks are practical

Se Eun Oh*, Shuai Li, and Nicholas Hopper

Fingerprinting Keywords in Search Queries over Tor

Timing Analysis of Keystrokes and Timing Attacks on SSH*

Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on fon-iks

Andrew M. White*  Austin R. Matthews†  Kevin Z. Snow*  Fabian Monrose*
*Department of Computer Science †Department of Linguistics
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina
{amw, kzsnow, fabian}@cs.unc.edu, armatthe@email.unc.edu

Abstract—In this work, we unveil new privacy threats against Voice-over-IP (VoIP) communications. Although prior work ciphers for encryption—interact to leak substantial information about a given conversation. Specifically, researchers...
Attack overview

• Predict search queries using only client traffic

• Combine multiple independent weak predictors
  • Escaped URL characters
  • HTTP2 header compression
  • Key-press time intervals
  • Natural language
Threat model

• Capture encrypted traffic at the NIC

• Victim types lowercase English letters + Space
  • No typos/backspace

• Autocomplete requests triggered by keydown events
Attack workflow

(Packet trace)

Keystroke detection

Tokenization

Dictionary pruning

Word identification

Beam search

The lazy dog
The lazy fox
And that dog
Autocomplete GET requests

GET /complete/search?q=t&cp=1
GET /complete/search?q=th&cp=2
GET /complete/search?q=the&cp=3
GET /complete/search?q=the%20&cp=4
GET /complete/search?q=the%201&cp=5
GET /complete/search?q=the%201a&cp=6
GET /complete/search?q=the%201az&cp=7
GET /complete/search?q=the%201azy&cp=8
Keystroke detection

- Find the longest increasing subsequence (LIS) of packet sizes

Baidu example:
searching for “the lazy dog”
Tokenization

GET /complete/search?q=t&cp=1
GET /complete/search?q=th&cp=2  +1
GET /complete/search?q=the&cp=3  +1
GET /complete/search?q=the%20&cp=4  +3
GET /complete/search?q=the%20l&cp=5  +1
GET /complete/search?q=the%20la&cp=6  +1
GET /complete/search?q=the%20la&cp=7  +1
GET /complete/search?q=the%20lazy&cp=8  +1
HPACK (HTTP2 header compression)

Static Huffman Encoding

```
0 1 2 3 4 5 6 7
+------------------+
| H | String Length (7+) |
+------------------+
| String Data (Length octets) |
```
PETAL
(Preset Encoding Table Information Leakage)

D O G S: $6 + 5 + 6 + 5 = 22$ bits

F I S H: $6 + 5 + 5 + 6 = 22$ bits

V O T E: $7 + 5 + 5 + 5 = 22$ bits
Incremental compression

One of these...

D O G S

D O G

a e i o s t

22 bits

17 bits

5 bits
Dictionary pruning

Observed
guns
dogs
Word identification

• Use a BiRNN to predict keys
Language model and beam search

Which word comes next?

> the lazy ____

1) dog
2) car
3) hat
4) big

Top 50 hypotheses

the lazy dog
the blue car
and some fox
... how they run
Data collection and results

• Data collect
  • Browser automation with Selenium
  • Replay keystrokes with uinput
  • 4k unique queries
  • 2 search engines (Google, Baidu)
  • 2 browsers (Chrome, Firefox)
  • 16k total queries recorded

• Keystroke detection and tokenization accuracy
  • > 99% (Google and Baidu)

• Top-50 classification accuracy (entire query is correct)
  • 15% (Google)
  • 13% (Baidu)

Example

Truth
he is recovering from a sprained

good hypotheses
he is recovering from a sprained
he is recovering from a strained

Bad hypotheses

to be president from a position
is to learn from such a position
Conclusions

• This attack has many of moving parts...
  • Several independent weak side channels combine to create a strong one

• Language modeling is key
  • The predictability of human behavior is difficult to mask

• Where else does incremental compression occur?
  • Thin clients/websites with autosave feature?
  • Mapping services (latitude/longitude changes incrementally)?
Thank you

• Source code
  kreae (keystroke recognition and entropy elimination program)
  https://github.com/vmonaco/kreae

• Contact me
  https://vmonaco.com

• Questions?