Phishing for a title

Anonymous authors

ABSTRACT

ACM Reference Format:
Anonymous authors. 2024. Phishing for a title. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (CCS24).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

2 BACKGROUND

2.1 Evolution of Phishing over time

2.2 PhishingKits as a service

2.3 Tracking phishing in the wild

3 METHODOLOGY

3.1 Phishing Feeds and crawling

3.2 Data post processing

3.3 Client-side kit detector

3.4 Analysis

3.4.1 Cloaking identification.

3.4.2 First Party / Third Party identification.

3.4.3 MDN APIs.

3.4.4 Kit identification.

4 RESULTS

4.1 Browser API trends

4.1.1 Fingerprinting APIs.

- The usage of seed APIs used by Su at el.[**fptechniques-www23**] (well know APIs that trackers use) has been steadily decreasing from 30% of the daily traffic to 15%.

4.1.2 MDN API groups.

- We collected **134** MDN API categories
- TODO: Compresison to real tranco pages
- TODO: Compresison to target pages
- HDBSCAN clustering of MDN API categories over time

CCS24, June 03–05, 2024, Woodstock, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Figure 1: Crawling infrastructure

Table 1: The usage of WASM related Browser APIs

Month	**Total WASM API calls**
August 2023	**61**
September 2023	**369**
October 2023	**337**
November 2023	**261**
December 2023	**350**
January 2024	**371**
February 2024	**302**
March 2024	**428**

4.1.3 Experimental APIs and WASM.

- We discover presese of 14 experimental MDN categories s
- TODO: Do these categories go up or down over time?
- We observe a constant number of WASM API calls over the span of the 6 months.
- We observed using WASM modules for client-side bot detection via captchas

4.1.4 Interactive JS logs.

-

4.1.5 Client-side cloaking tactics.

- Cloaking APIs
- APIs that are conditioned on
- User interaction API popularity

4.1.6 First Party/Third Party embedded/Third Party Scripts.

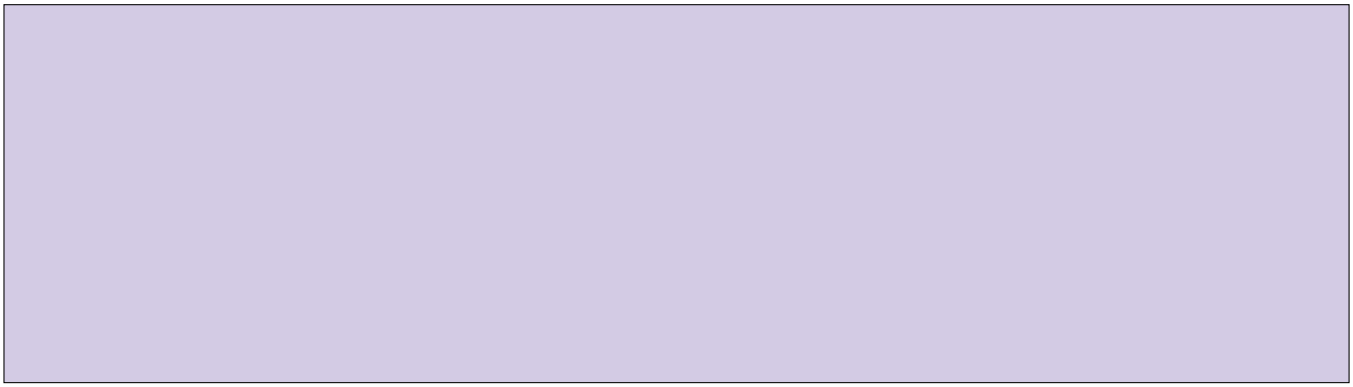

Figure 2: Server-side and Client-side cloaking over time

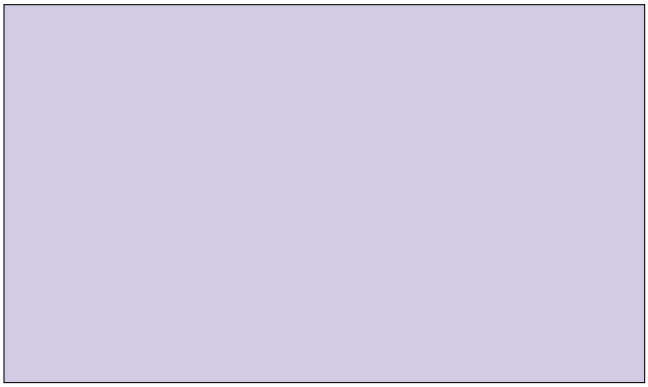

Figure 3: Crypto.getRandomValues API call over time

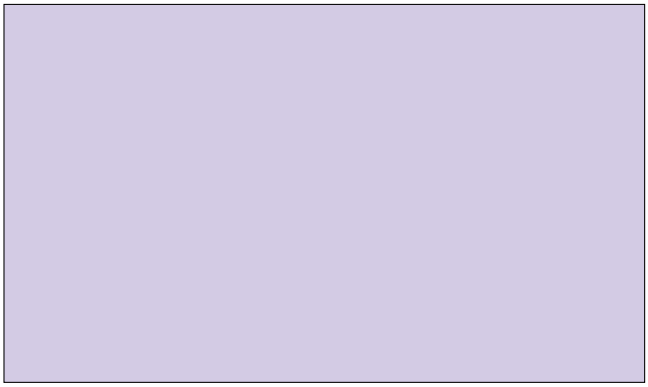

Figure 4: Seed FP APIs vs all FP APIs discovered by Su *et al.*

4.2 Kit families

- Overview of how many kits, how many de-duped with hash, how many de-depued with jaccard

4.2.1 Offline evaluation of detector.

- We evaluate the detector on 70% of the domains that have a phishing kit attached
- We evaluate it on accuracy and ability to distinguish new kits. TODO THIS IS WHAT WE NEED TO DO

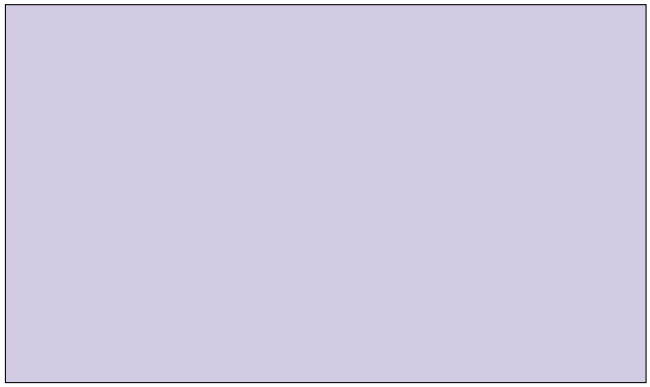

Figure 5: Clusters of seen kits

4.2.2 Kit fragments and inheritence.

- Using the SHA hash of scripts, we can detect scripts that some pages emmbed as first party while other load from a third party source.

4.3 Kit detection

4.3.1 Kit detection via javascript.

-

5 CASE STUDIES

5.1 Open Source client components

- Git directories in phishing kits
- Finding OSS components in client-side javascript via Github API

5.2 Mobile targets

- MDN distribution v. other sources
- APIs specific to mobile devices

5.3 Comparison to target pages

-

6 RELATED WORK

Measurement studies of phishing sites: Recent work in analizing phishing websites from phishing feeds includes Crawlphish [3], which uses a modified webkit engine to force execute all the paths in client-side javascript to identify cloaking behavior, Rods with Laser Beams [2] which uses a extension to capture set list of fingerprinting APIs, and identified the user of third party fingerprinting scripts in phishing pages that are not of the original target page, and Catching Phishers By Their Bait [1], which studied and identified phishing kits via manually crafted DOM and Javascript fingerprints.

Phishing kit analysis: Prior work has examined phishing kits to identify tradecraft within the ecosystem , studying similarity of kits observed , or in order to identify pages targetting a paticular theme.

Phishing page identification: Everyone under the sun has thrown ML at this problem, some state of the art work has looked at constructing knowlage based detection tools, and ofc, someome has done LLM To our knowlage, we are the first paper to automatically isolate, and craft detection fingerprints, and we are first to do it at the level that VV8 lets us do it.

7 LIMITATIONS AND FUTURE WORK

7.1 Obfuscation and Flow analysis

7.2 Automated submissions

8 CONCLUSION

REFERENCES

[1] Hugo Bijmans, Tim Booij, Anneke Schwedersky, Aria Nedgabat, and Rolf van Wegberg. 2021. Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection. In 30th USENIX Security Symposium (USENIX Security 21), 3757–3774. ISBN: 978-1-939133-24-3. Retrieved Mar. 12, 2024 from https://www.usenix.org/conference/usenixsecurity21/presentation/bijmans.

[2] Iskander Sanchez-Rola, Leyla Bilge, Davide Balzarotti, Armin Buescher, and Petros Efstathopoulos. 2023. Rods with Laser Beams: Understanding Browser Fingerprinting on Phishing Pages. In 32nd USENIX Security Symposium (USENIX Security 23), 4157–4173. ISBN: 978-1-939133-37-3. Retrieved Mar. 12, 2024 from https://www.usenix.org/conference/usenixsecurity23/presentation/sanchez-rola.

[3] Penghui Zhang et al. 2021. Crawlphish: large-scale analysis of client-side cloaking techniques in phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*, 1109–1124. DOI: 10.1109/SP40001.2021.00021.