PhishLang

A Real-Time, Fully Client-Side Phishing Detection Framework Using MobileBERT

Sayak Saha Roy, *Shirin Nilizadeh*

Read the paper here

NDSS SYMPOSIUM

2026 and Phishing is still a Massive Problem

Phishing is identified as the *most commonly used tactic* in cybercrime.

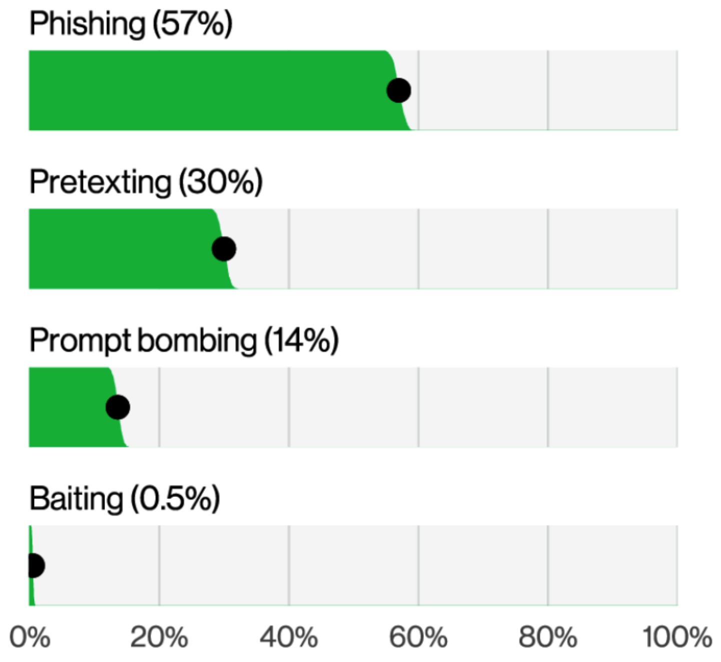
2 million

unique phishing attacks*

recorded between May 2024 and April 2025.

423% growth

Increase since 2020*



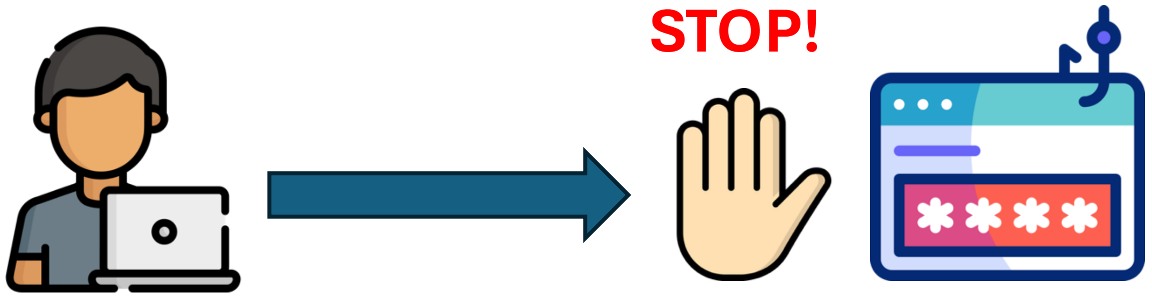
Top select Social action varieties in Social Engineering incidents (n=3,208)

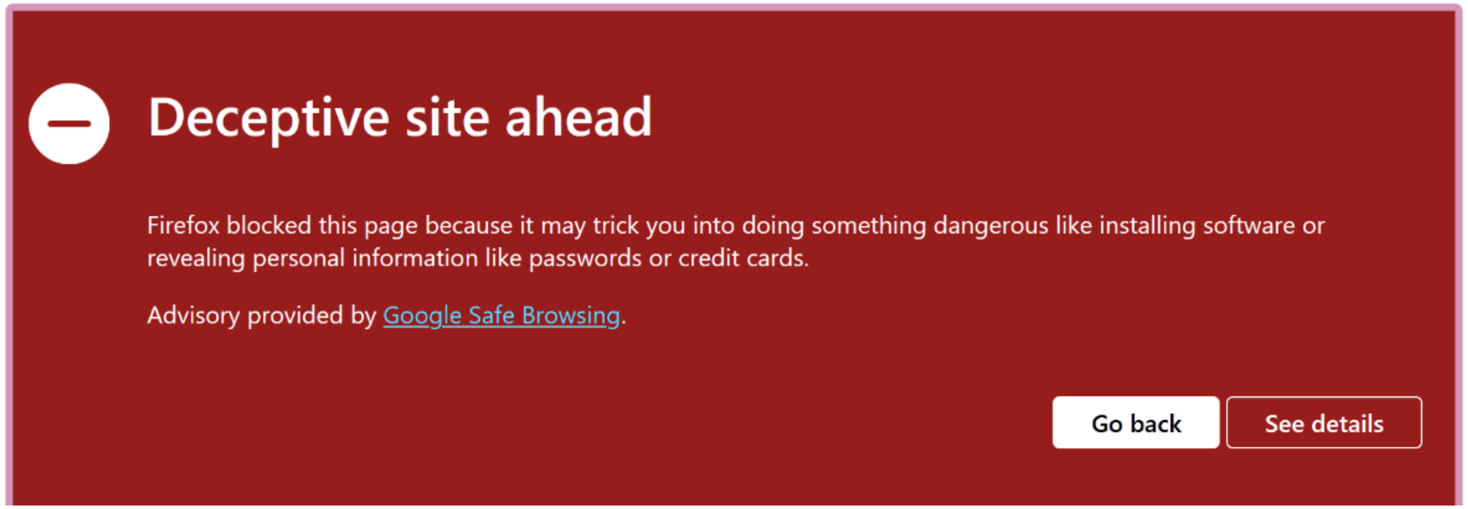
https://docs.apwg.org/reports/apwg_trends_report_q2_2025.pdf

https://www.verizon.com/business/resources/reports/2025-dbir-data-breach-investigations-report.pdf

Detection = Prevention

Phishing Website Detectors




Deceptive site ahead

Firefox blocked this page because it may trick you into doing something dangerous like installing software or revealing personal information like passwords or credit cards.

Advisory provided by Google Safe Browsing.

The Core Challenge

Server-side detection to populate Blocklists

Security vendors run Phishing Detectors run on millions of URLs daily.
Despite being capable, new attacks might have a detection buffer.

Client-side detection – On-Device

Phishing Detectors exists on device.
Evaluation of websites happens on-demand, no buffer.
Severely resource constrained
Simpler/Bare-bones models implemented compared to Server-side

Phishing Website Detectors (PWDs)

Feature based detection:

Static features in URL/Domains and HTML source:

Brand recognition, Code characteristics, URL Patterns, TLD, Domain age etc.

Fast, heavily used by blocklists and vendors

Overfitted to hand-crafted features.

Fail against New/evasive scams

Phishing Website Detectors (PWDs)

Behavioral detectors:

Visual similarity, Dynamic page behaviour etc.

+ High Efficiency, even against evasive threats

- Significant resource demand

- Often Slow

Impractical to use on production or client-side systems

What about Large Language Models (LLMs) ?

ChatGPT Claude LLaMA Gemma

Have been steadily used in phishing detection

Commercial LLMs:

+ Strong performance (98% accuracy)*

- API-dependent = Expensive at Scale

What about Large Language Models (LLMs) ?

Have been steadily used in phishing detection

Open-source LLMs

Large Models: GPU and memory intensive, and slow.

- Impracticle to run at scale or client-side.

Small Models:

Fine-tuned on static feature sets,

Same issues as feature-based models.

Language Models (LMs) have a huge advantage

Can understand semantic context

Language models capture relationships between tokens and structural elements to model intent rather than surface patterns.

Already has been applied for source code analysis[1], malware detection[2,3].

For websites, this means

They can identify how different elements inside a webpage interact with each other to express behavioral intent.

[1] https://ieeexplore.ieee.org/abstract/document/9463129
[2] https://www.mdpi.com/2504-2289/7/2/60, [3] https://ieeexplore.ieee.org/abstract/document/9486377

Key Motivation

A language model-based phishing detector that is:

1) feature-agnostic,
2) lightweight, fast, and highly accurate.
3) Can be run on-device

Groundtruth

PhishPedia [Lin at al[1],[2]]

- 30K Phishing (Source from OpenPhish) + 30K Benign website artifacts
- Widely used and validated in public research.
- Contains source code, screenshots.
- **22, 419 Phishing + 26,000 Benign that had HTML source code**

Final Groundtruth

[1] https://www.usenix.org/conference/usenixsecurity21/presentation/lin
[2] https://sites.google.com/view/phishpedia-site/home

Key challenge

Using a small Language Model

- Low token limit, smaller context windows

We have to reduce the input that we provide to the model.

The model needs to learn from the most **informative and behaviorally relevant** areas of the website that signal phishing intent.

Identifying website tags

Two coders manually evaluated 500 (randomly selected) phishing websites from the Ground truth

Goal: Identify general purpose <tags> that contribute most to the behavior of the website (high-signal).

Identifying website tags

Findings:

- Title (title) and Headings (h1,h2,h3..) - Capture user attention with alarming or misleading statements
- Links (a) – Often empty/placeholder/incriminating text
- Paragraphs (p) - Persuasive Narratives
- List tags (ul,ol) – Structured information
- Forms (form) - Capturing sensitive user data,
- Input fields (input) – Often inside forms.
- Buttons (button) – Coupled with forms
- Scripts (script) and iframe – Embedded context

Preparing Training Dataset

Small Language models train best on content that has structural consistency.

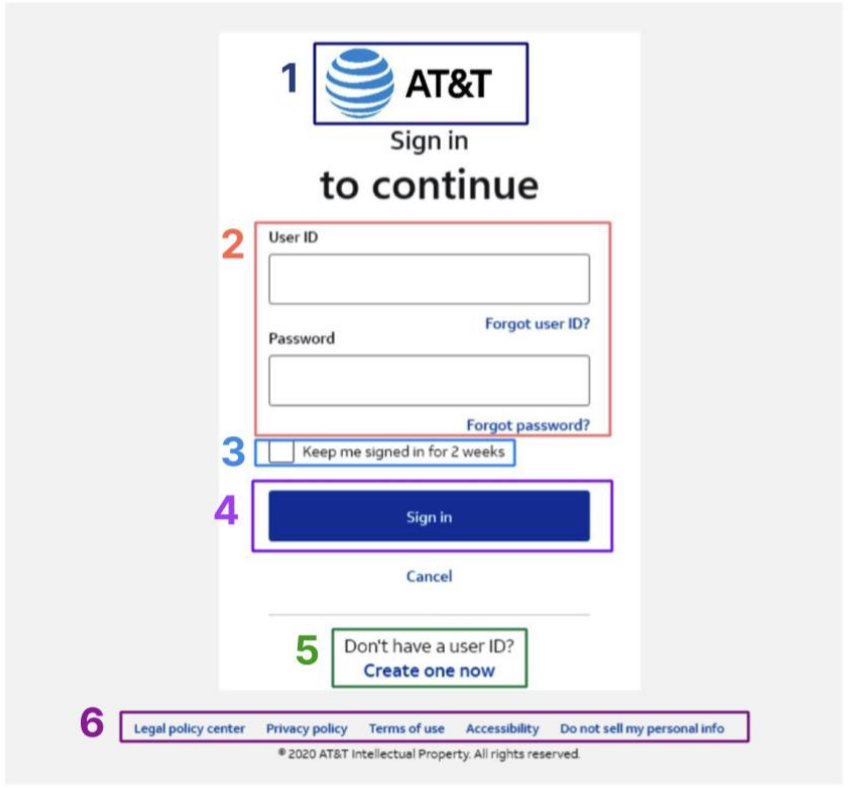
Parse actionable HTML tags (with content) from **Groundtruth** for both Phishing and Benign websites.

Given website’s source code, extract the textual content and selected attributes of each tag, and convert them into a structured representation.

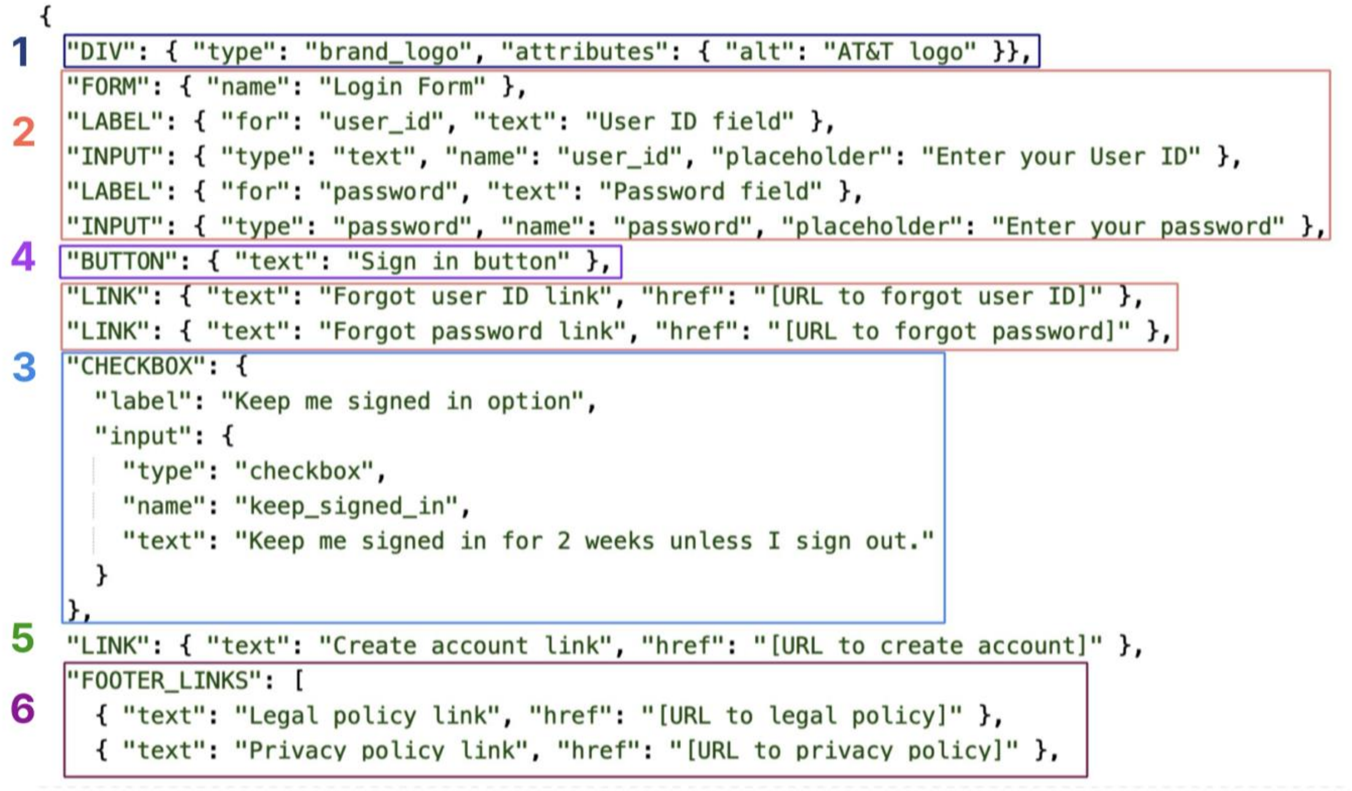
Each element is represented by its tag type, followed by its content and relevant attributes.

Example of a parsed representation

Original website → Parsed structure



Original website



```
  {
1 "DIV": { "type": "brand_logo", "attributes": { "alt": "AT&T logo" }},
  "FORM": { "name": "Login Form" },
2 "LABEL": { "for": "user_id", "text": "User ID field" },
  "INPUT": { "type": "text", "name": "user_id", "placeholder": "Enter your User ID" },
  "LABEL": { "for": "password", "text": "Password field" },
  "INPUT": { "type": "password", "name": "password", "placeholder": "Enter your password" },
4 "BUTTON": { "text": "Sign in button" },
  "LINK": { "text": "Forgot user ID link", "href": "[URL to forgot user ID]" },
  "LINK": { "text": "Forgot password link", "href": "[URL to forgot password]" },
3 "CHECKBOX": {
    "label": "Keep me signed in option",
    "input": {
      "type": "checkbox",
      "name": "keep_signed_in",
      "text": "Keep me signed in for 2 weeks unless I sign out."
    }
  },
5 "LINK": { "text": "Create account link", "href": "[URL to create account]" },
6 "FOOTER_LINKS": [
    { "text": "Legal policy link", "href": "[URL to legal policy]" },
    { "text": "Privacy policy link", "href": "[URL to privacy policy]" },
```

Parsed representation

Training approach

Train a language model on the parsed representation

✗ **This tag contains phishing content.**

✓ **Given that these tags exist together in this page, is the context provided by them together consistent with a phishing or a benign website?**

Choosing the best model

We trained and tested 9 Language models to find the best fit for

Reliability (Correctness)

Memory usage

Speed (Time)

Model Selection

MobileBERT provides the best trade-off

Best performance but not chosen

Model	Accuracy	Precision	Recall	F1 Score	Time (s)	Memory
MobileBERT	**0.96**	**0.95**	**0.96**	**0.96**	**0.39**	**74MB**
DistilBERT	0.94	0.94	0.94	0.94	0.85	502MB
DeBERTa	0.83	0.83	0.84	0.84	1.02	1,341MB
FastText	0.62	0.65	0.63	0.63	1.43	201MB
GPT-2	0.68	0.64	0.69	0.68	1.81	922MB
TinyBERT	0.85	0.88	0.82	0.84	0.78	495MB
Llama2 (7B)	0.97	0.96	0.96	0.96	33.71	4,873MB
T5-base	0.89	0.88	0.88	0.88	12.40	1,279MB
Bloom (560M)	0.75	0.74	0.74	0.74	7.49	7,352MB

Comparison with other PWDs

Comparing our MobileBERT model with 4 open-source SOTA models

Visual PhishNet, PhishIntention (Image based)
StackModel (URL + HTML)
URLNet (URL only)

Also, ChatGPT 3.5T and GPT 4:

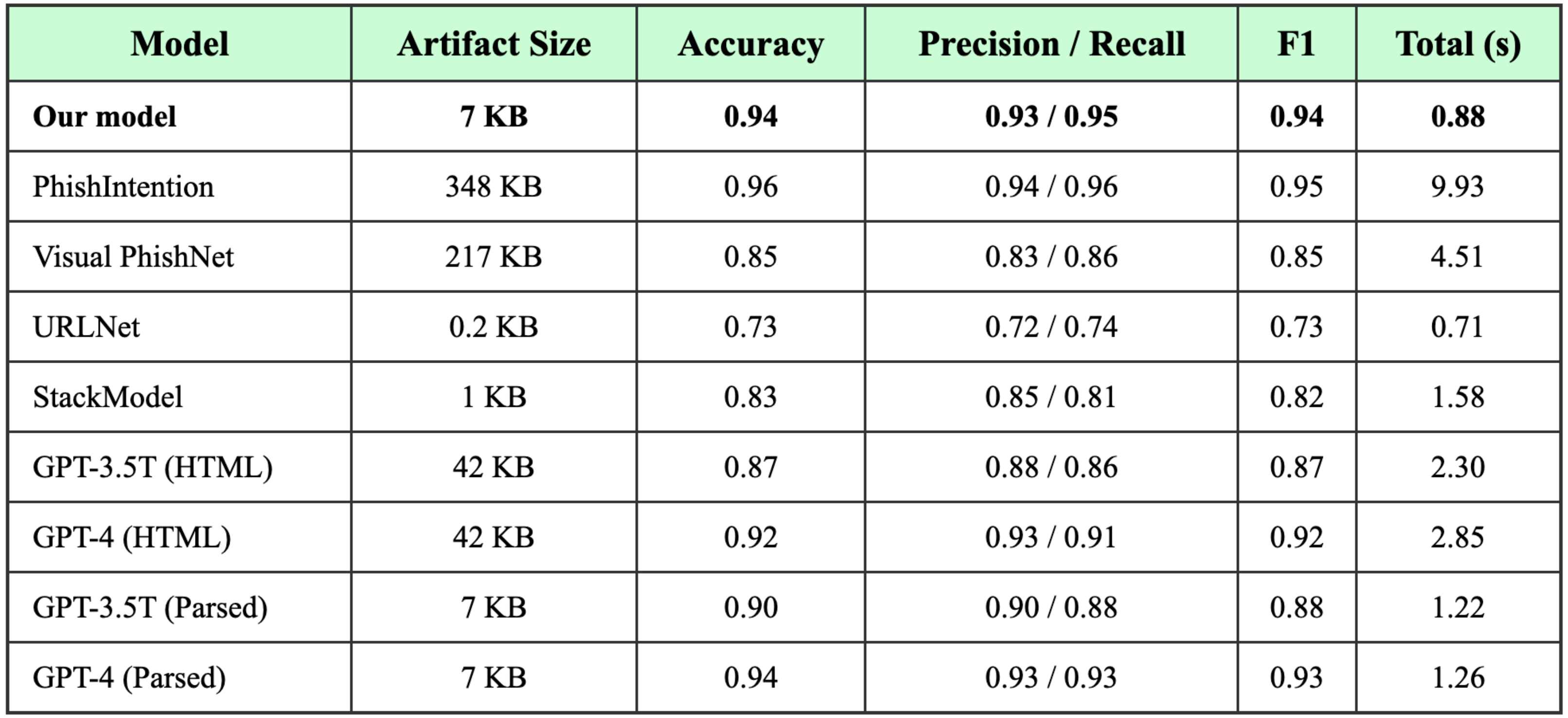
- Full HTML
- Our parsed version

Comparison with other PWDs

Our model is **atleast 10.2x faster** than the best performing Visual models while **requiring 30k less storage.**

Loses **~1% of the effeciency**

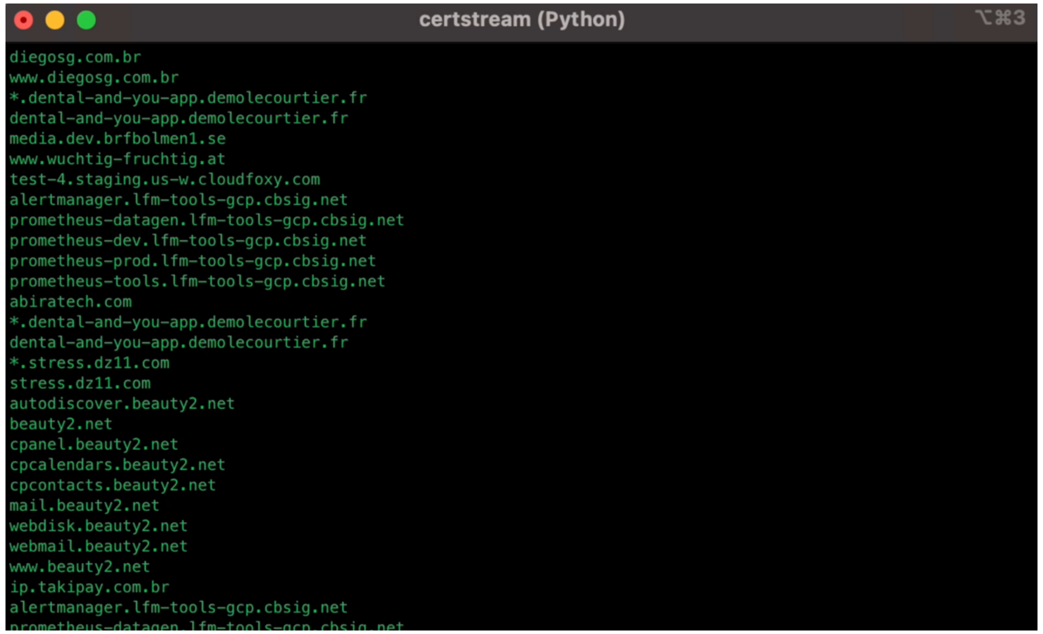
Performs better than ChatGPT 3.5T and is on par with GPT 4.



Model	Artifact Size	Accuracy	Precision / Recall	F1	Total (s)
Our model	**7 KB**	**0.94**	**0.93 / 0.95**	**0.94**	**0.88**
PhishIntention	348 KB	0.96	0.94 / 0.96	0.95	9.93
Visual PhishNet	217 KB	0.85	0.83 / 0.86	0.85	4.51
URLNet	0.2 KB	0.73	0.72 / 0.74	0.73	0.71
StackModel	1 KB	0.83	0.85 / 0.81	0.82	1.58
GPT-3.5T (HTML)	42 KB	0.87	0.88 / 0.86	0.87	2.30
GPT-4 (HTML)	42 KB	0.92	0.93 / 0.91	0.92	2.85
GPT-3.5T (Parsed)	7 KB	0.90	0.90 / 0.88	0.88	1.22
GPT-4 (Parsed)	7 KB	0.94	0.93 / 0.93	0.93	1.26

Real-world deployment

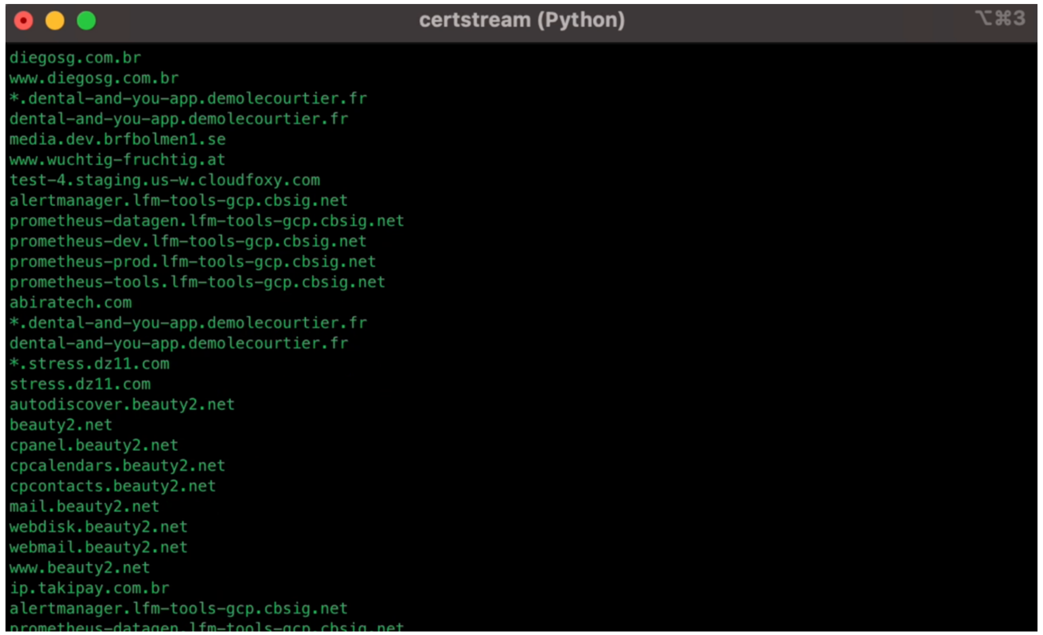
Running PhishLang over Certstream for there months



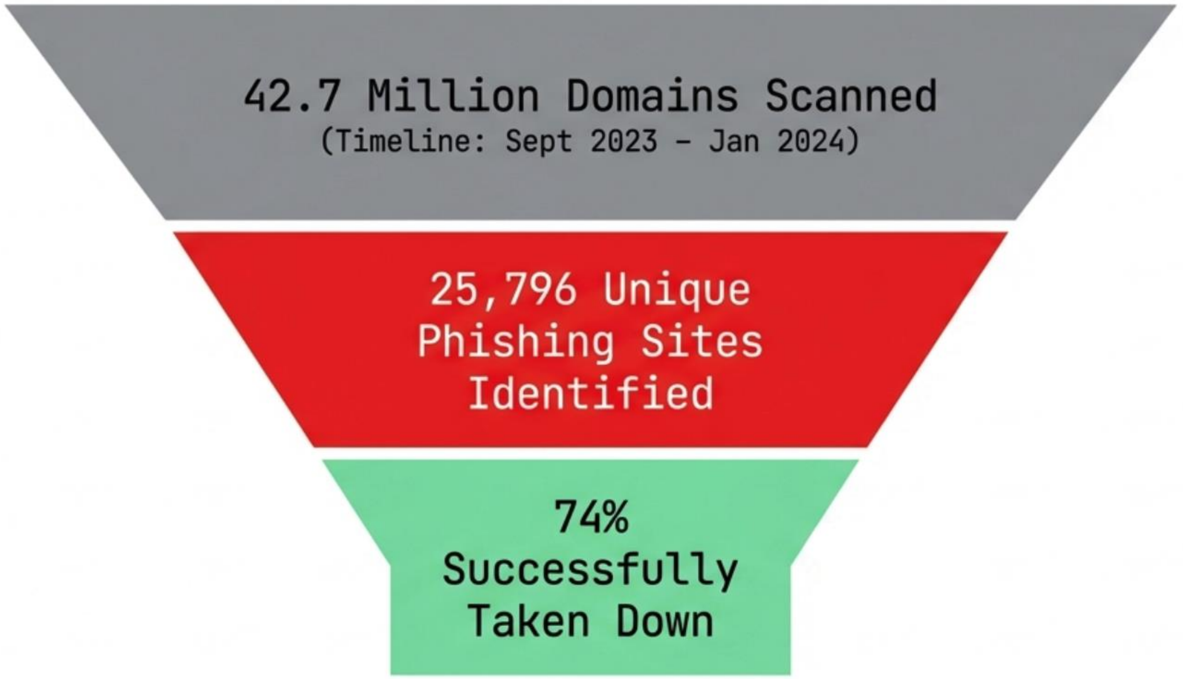
```
diegosg.com.br
www.diegosg.com.br
*.dental-and-you-app.demolecourtier.fr
dental-and-you-app.demolecourtier.fr
media.dev.brfbolmen1.se
www.wuchtig-fruchtig.at
test-4.staging.us-w.cloudfoxy.com
alertmanager.lfm-tools-gcp.cbsig.net
prometheus-datagen.lfm-tools-gcp.cbsig.net
prometheus-dev.lfm-tools-gcp.cbsig.net
prometheus-prod.lfm-tools-gcp.cbsig.net
prometheus-tools.lfm-tools-gcp.cbsig.net
abiratech.com
*.dental-and-you-app.demolecourtier.fr
dental-and-you-app.demolecourtier.fr
*.stress.dz11.com
stress.dz11.com
autodiscover.beauty2.net
beauty2.net
cpanel.beauty2.net
cpcalendars.beauty2.net
cpcontacts.beauty2.net
mail.beauty2.net
webdisk.beauty2.net
webmail.beauty2.net
www.beauty2.net
ip.takipay.com.br
alertmanager.lfm-tools-gcp.cbsig.net
prometheus-datagen.lfm-tools-gcp.cbsig.net
```

Real-world deployment

Running PhishLang over Certstream for there months



```
diegosg.com.br
www.diegosg.com.br
*.dental-and-you-app.demolecourtier.fr
dental-and-you-app.demolecourtier.fr
media.dev.brfbolmen1.se
www.wuchtig-fruchtig.at
test-4.staging.us-w.cloudfoxy.com
alertmanager.lfm-tools-gcp.cbsig.net
prometheus-datagen.lfm-tools-gcp.cbsig.net
prometheus-dev.lfm-tools-gcp.cbsig.net
prometheus-prod.lfm-tools-gcp.cbsig.net
prometheus-tools.lfm-tools-gcp.cbsig.net
abiratech.com
*.dental-and-you-app.demolecourtier.fr
dental-and-you-app.demolecourtier.fr
*.stress.dz11.com
stress.dz11.com
autodiscover.beauty2.net
beauty2.net
cpanel.beauty2.net
cpcalendars.beauty2.net
cpcontacts.beauty2.net
mail.beauty2.net
webdisk.beauty2.net
webmail.beauty2.net
www.beauty2.net
ip.takipay.com.br
alertmanager.lfm-tools-gcp.cbsig.net
prometheus-datagen.lfm-tools-gcp.cbsig.net
```


42.7 Million Domains Scanned
(Timeline: Sept 2023 – Jan 2024)
25,796 Unique Phishing Sites Identified
74% Successfully Taken Down

Detections reported to Domain Registrars and Anti-phishing blocklists

Validating detections

Randomly sampled 2,500 detections.
Manually evaluated 2,388 to be True positives (95.5%)

Identifying evasive phishing detection

Utilized 18 features from prior literature towards detecting:

- Regular Phishing **(1,623)**
- Behavioral JS Evasion **(280)**
- Clickjacking **(313)**
- Dom Manipulation **(94)**
- Text encoding **(78)**

Validating detections

Randomly sampled 2,500 detections.
Manually evaluated 2,388 to be True positives (95.5%)

Identifying evasive phishing detection

Utilized 18 features from prior literature towards detecting:

- Regular Phishing **(17,396 – 75%)**
- Behavioral JS Evasion **(3,159 – 13%)**
- Clickjacking **(3,349)**
- Dom Manipulation **(1,057)**
- Text encoding **(835)**

Extrapolated features automatically across the entire detection set

Blocklist Performance

Websites reported to four blocklists

Google Safe Browsing Microsoft PhishTank OpenPhish

What % of URLs reported by PhishLang were already detected?

Blocklist Performance

Reported URLs that were already detected

	Google Safe Browsing	Microsoft	PhishTank	OpenPhish
Regular Attack	50.92%	34.26%	14.8%	22.6%
JS Evasion	35.27%	27.48%	3.9%	8.5%
ReCAPTCHA/QR	28.6%	23.1%	4.4%	12.1%
Clickjacking	31.1%	26.2%	5.5%	14.2%
DOM Attacks	22.5%	18.6%	3.9%	9.8%
Text Encoding	19.8%	15.4%	2.5%	7.4%

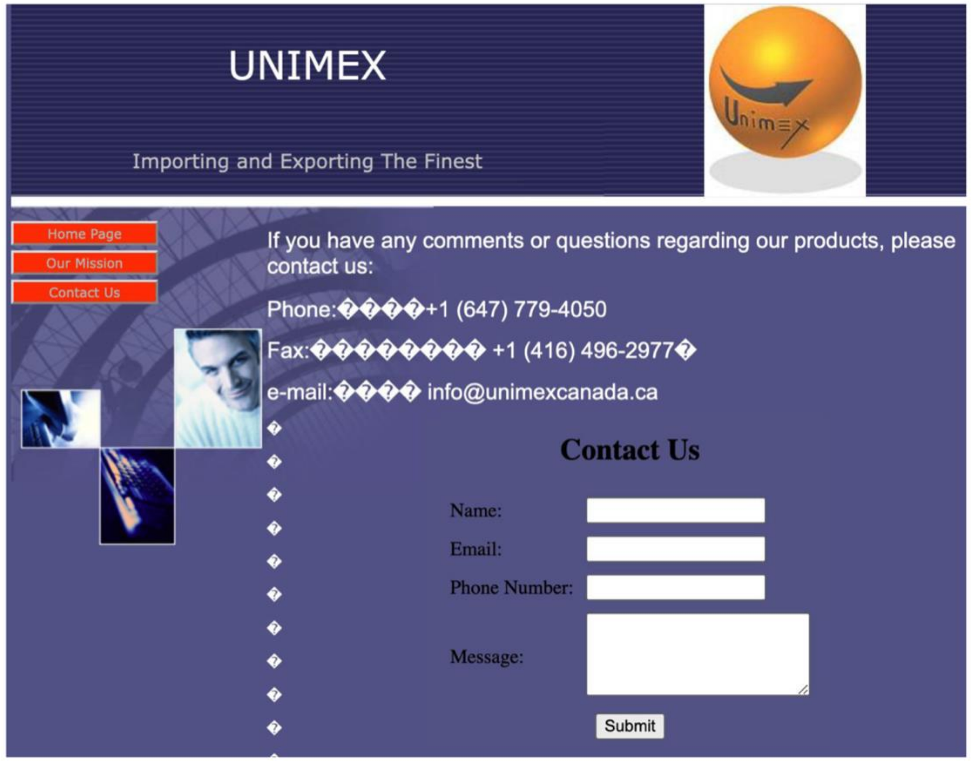
Low (0%) High (100%)

Takeaway: PhishLang can help anti-phishing measures identify new phishing sites.

Misdetections – False Positives

Malformed / Poorly Structured Websites

- Poor HTML structure and invalid characters
- Too many Input fields requesting personal information
- Empty links which do not go anywhere.
- Empty page with only login fields



UNIMEX

Importing and Exporting The Finest

If you have any comments or questions regarding our products, please contact us:

Phone:����+1 (647) 779-4050

Fax:�������� +1 (416) 496-2977�

e-mail:���� info@unimexcanada.ca

�
�
�
�
�
�
�
�
�
�

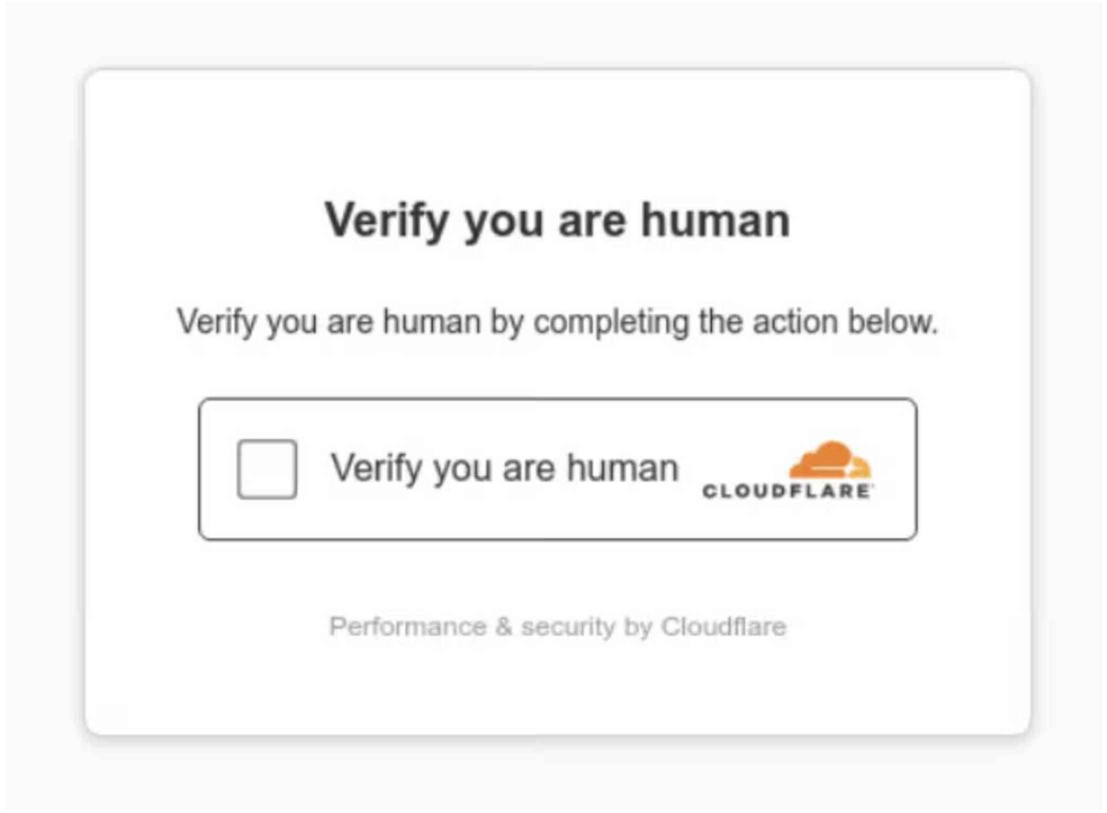
Contact Us

Name:	
Email:	
Phone Number:	
Message:	

Submit

Misdetections – False Negatives

Captcha Phishing:
- No fix



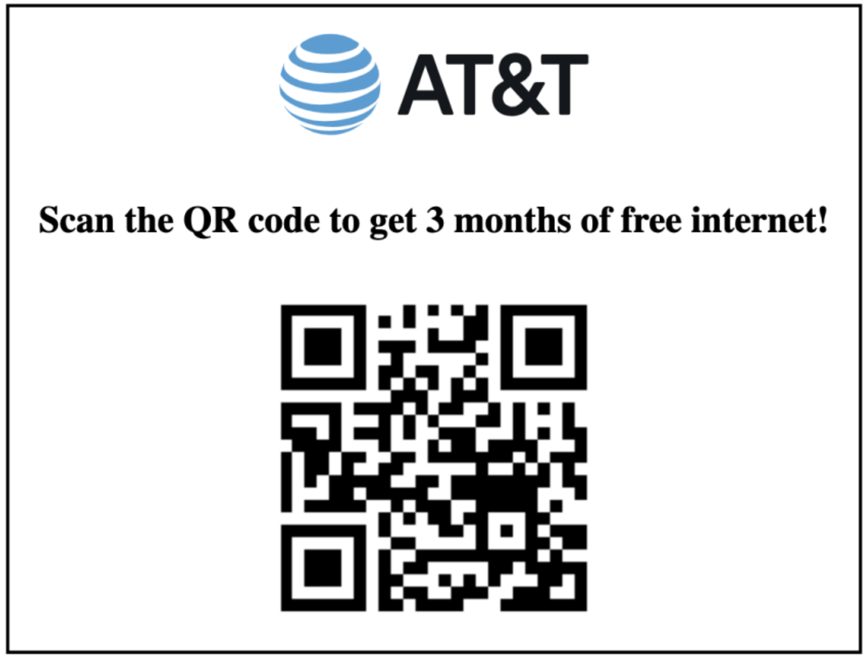
Verify you are human

Verify you are human by completing the action below.

Verify you are human

Performance & security by Cloudflare

QR-Code Phishing:
- Implemented *pyzbar* to automatically scan QR

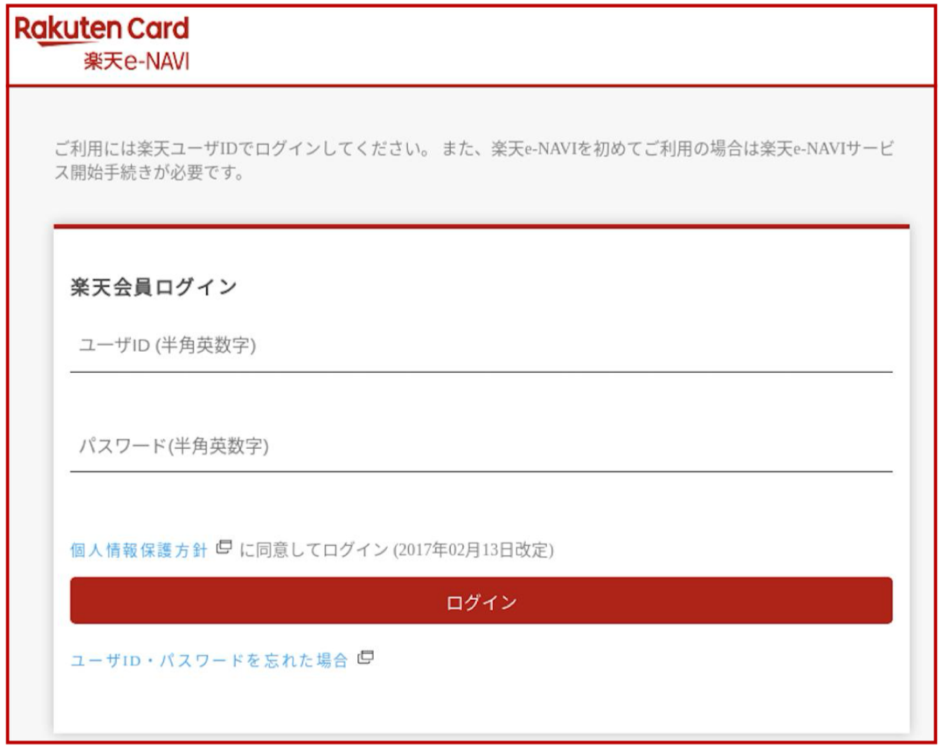


AT&T

Scan the QR code to get 3 months of free internet!

Misdetections – False Negatives

Non-English Phishing:



Rakuten Card
楽天e-NAVI

ご利用には楽天ユーザIDでログインしてください。 また、楽天e-NAVIを初めてご利用の場合は楽天e-NAVIサービス開始手続きが必要です。

楽天会員ログイン

ユーザID (半角英数字)

パスワード(半角英数字)

個人情報保護方針 に同意してログイン (2017年02月13日改定)

ログイン

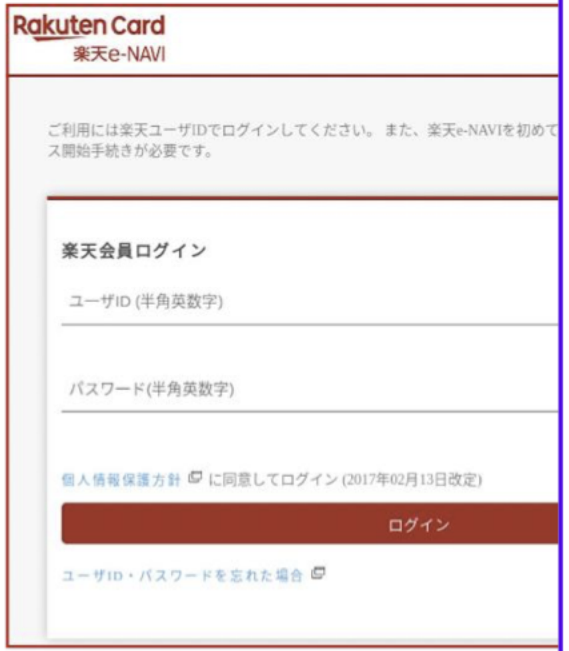
ユーザID・パスワードを忘れた場合

Misdetections – False Negatives

Non-English Phishing:

Langdetect + Argos Translate

Translated, parsed and detected



Rakuten Card
楽天e-NAVI

ご利用には楽天ユーザIDでログインしてください。 また、楽天e-NAVIを初めて
ス開始手続きが必要です。

楽天会員ログイン

ユーザID (半角英数字)

パスワード(半角英数字)

個人情報保護方針 に同意してログイン (2017年02月13日改定)

ログイン

ユーザID・パスワードを忘れた場合

Original Site

```
"FORM": {
  "title": "Rakuten Member Login",

  "TEXT": "To use this service, please log in with your Rakuten user ID.
If this is your first time using Rakuten e-NAVI, you must complete the e-NAVI service start
procedure.",

  "INPUT": [
    { "type": "text", "name": "user_id"},
    { "type": "password", "name": "password", "placeholder": "Password (alphanumeric)" }
  ],

  "CHECKBOX": {
    "label": "Agree to Privacy Policy",
  },

  "BUTTON": {
    "text": "Login"
  },

  "LINK": [
    {
      "text": "Forgot your user ID or password?",
      "href": "[URL to recovery page]"
    },
    {
      "text": "Privacy Policy",
      "href": "[URL to privacy policy]"
    }
  ]
```

Adversarial Robustness

Perturbations can reduce confidence of the model = Misclassifications

Prior work

Image space (pixel-level FGSM attacks)
URL feature space manipulation

– May introduce visible artifacts
Model specific
- Assumes attacker has access to model internals

Our approach

- Real attackers operate in the **problem space**

  Adopted 15 realistic evasive attacks (Yuan et al.[1])
  - Direct HTML manipulations that:
  – Preserve visual appearance
  – Preserve phishing functionality
  – Modify tag content and structure

Adversarial Robustness

Broadly, 4 types of attacks

1. Link Injection

Goal: Distort internal–external link statistics

InjectIntElem (A1), InjectIntElemFooter (A2), InjectIntLinkElem (A3), InjectExtElem (A4), InjectExtElemFooter (A5), UpdateIntAnchors (A10)

2. Hidden Element Manipulation

Goal: Add structural noise without affecting rendering

UpdateHiddenDivs (A11), UpdateHiddenButtons (A12), UpdateHiddenInputs (A13), UpdateIFrames (A15), InjectFakeCopyright (A9)

Adversarial Robustness

Broadly, 4 types of attacks

3. Form and Action Manipulation

Goal: Alter credential submission behavior

UpdateForm (A6) – Replace <form action> with benign internal link
ObfuscateExtLinks (A7) - Replace external links, restore via JavaScript at runtime

4. Script / Content Obfuscation

Goal: Hide malicious behavior in scripts

ObfuscateJS (A8) – Base64-encode JavaScript and decode at runtime
UpdateTitle (A14) - Dynamically alter <title> to evade text cues

Adversarial Robustness

Not all attacks will be appropriate for each phishing sample

Use an optimizer (Montaruli et al[1]) to find the best attack for each sample using - **Single round** or **Multi-round** manipulations.

All 22,419 ground-truth samples perturbed and evaluated using PhishLang.

Two metrics:

Adversarial advantage – Reduction of model confidence
Incorrect detections – Misdetection of adversarial samples

[1] https://dl.acm.org/doi/abs/10.1145/3605764.3623920

Adversarial Robustness

Link Injection / Ratio Manipulation

Attack	Median advantage	Incorrectly predicted
InjectIntElem (A1)	0.204	7.99%
InjectIntElemFoot (A2)	0.142	11.24%
InjectIntLinkElem (A3)	0.231	4.43%
InjectExtElem (A4)	0.285	7.02%
InjectExtElemFoot (A5)	0.194	15.01%
UpdateIntAnchors (A10)	0.245	7.92%

Hidden Element Manipulation

Attack	Median advantage	Incorrectly predicted
UpdateHiddenDivs (A11)	0.173	8.01%
UpdateHiddenButtons (A12)	0.241	17.20%
UpdateHiddenInputs (A13)	0.403	13.59%
UpdateIFrames (A15)	0.111	5.42%
InjectFakeCopyright (A9)	0.201	4.10%

Form / Link Action Manipulation

Attack	Median advantage	Incorrectly predicted
UpdateForm (A6)	0.271	4.90%
ObfuscateExtLinks (A7)	0.314	19.01%

Script / Content Obfuscation

Attack	Median advantage	Incorrectly predicted
ObfuscateJS (A8)	0.348	20.92%
UpdateTitle (A14)	0.107	11.27%

Regular PWDs incorrectly predict 30-40% adversarial samples

PhishLang is already more robust.

Mitigating Adversarial Attacks

Two approaches:

Parser modification – Fix the vulnerability when creating the parsed representation
Adversarial retraining – Retrain PhishLang using Adversarial samples

Mitigating Adversarial Attacks

Six patches

Patch 1: Ignore tags with hidden attribute or display:none in inline styles. Fixes InjectIntElem (A1), InjectIntElemFooter (A2), InjectIntLinkElem (A3), InjectExtElem (A4), InjectExtElemFooter (A5)

Patch 1: Parse <style> blocks and ignore elements hidden via CSS display:none.

Parser Modification

Patch	Description	Fixes
Patch 1	Ignore tags with hidden attribute or display:none in inline styles.	InjectIntElem (A1), InjectIntElemFooter (A2), InjectIntLinkElem (A3), InjectExtElem (A4), InjectExtElemFooter (A5)
Patch 2	Parse <style> blocks and ignore elements hidden via CSS display:none.	InjectIntElem family (when using CSS hiding strategy S3)
Patch 3	Validate <form action> and external link destinations.	UpdateForm (A6), ObfuscateExtLinks (A7), UpdateIntAnchors (A10)
Patch 4.1	Decode Base64 or non-UTF-8 scripts before parsing.	ObfuscateJS (A8)
Patch 4.2	Adversarial retraining for JS obfuscation.	ObfuscateJS (A8)
Patch 5	Adversarial retraining for hidden button manipulation.	UpdateHiddenButtons (A12)
Patch 6	Adversarial retraining for hidden input manipulation.	UpdateHiddenInputs (A13) **Variant also used for:** UpdateTitle (A14)

Impact

Link Injection / Ratio Manipulation

Attack	Median advantage	Incorrectly predicted
InjectIntElem (A1)	0.204 **(0)**	7.99% **(0)**
InjectIntElemFoot (A2)	0.142 **(0)**	11.24% **(0)**
InjectIntLinkElem (A3)	0.231 **(0)**	4.43% **(0)**
InjectExtElem (A4)	0.285 **(0)**	7.02% **(0)**
InjectExtElemFoot (A5)	0.194 **(0)**	15.01% **(0)**
UpdateIntAnchors (A10)	0.245 **(0)**	7.92% **(0)**

Hidden Element Manipulation

Attack	Median advantage	Incorrectly predicted
UpdateHiddenDivs (A11)	0.173 **(0)**	8.01% **(0)**
UpdateHiddenButtons (A12)	0.241 **(0.025)**	17.20% **(1.07%)**
UpdateHiddenInputs (A13)	0.403 **(0.027)**	13.59% **(1.18%)**
UpdateIFrames (A15)	0.111 **(0)**	5.42% **(0)**
InjectFakeCopyright (A9)	0.201 **(0)**	4.10% **(0)**

Form / Link Action Manipulation

Attack	Median advantage	Incorrectly predicted
UpdateForm (A6)	0.271 **(0)**	4.90% **(0)**
ObfuscateExtLinks (A7)	0.314 **(0.039)**	19.01% **(1.68%)**

Script / Content Obfuscation

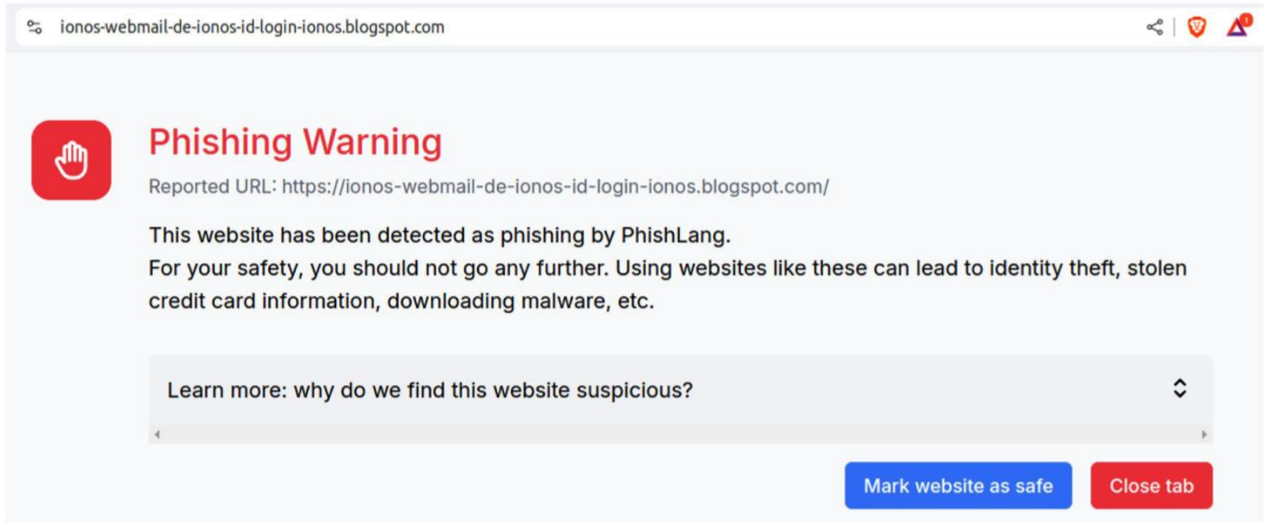
Attack	Median advantage	Incorrectly predicted
ObfuscateJS (A8)	0.348 **(0.028)**	20.92% **(1.51%)**
UpdateTitle (A14)	0.107 **(0.005)**	11.27% **(0.63%)**

Nullified several attacks and also reduced adversarial advantage.

Client-Side Application

PhishLang is memory efficient and fast.

First open-source fully client-side protection:
- No blocklist reliance
- Local MobileBERT server and browser extension.
- All inferences performed locally. Privacy focused.


ionos-webmail-de-ionos-id-login-ionos.blogspot.com

Phishing Warning

Reported URL: https://ionos-webmail-de-ionos-id-login-ionos.blogspot.com/

This website has been detected as phishing by PhishLang.
For your safety, you should not go any further. Using websites like these can lead to identity theft, stolen credit card information, downloading malware, etc.

Learn more: why do we find this website suspicious?

Evaluation

Ofcourse client-side PhishLang can run on high-end systems..

It should run on Low-end systems

Low-end system: Intel Celeron N4500 processor, 2GB of RAM, no GPU

Tested on 2,000 new phishing websites

89MB

Median RAM consumed

0.73s

Median inference time

Comparison with Commercial Tools

Real-time Zero-day protection

Hosted 500 new phishing domains using GoPhish.

- 100 regular phishing
- 100 each for four evasive categories
- Websites were ethically disabled (i.e. no credentials collected).
- Sites removed after experiment.

Comparison with Commercial Tools

Tested against

Two configurations:

1) Enhanced, 2) Client-side only

McAfee™ Bitdefender Avast TREND MICRO™

Overall

PhishLang – 457/500 (91.4%)

44.2%

Enhanced

15.4%*

Client-side

McAfee™ 45.4%

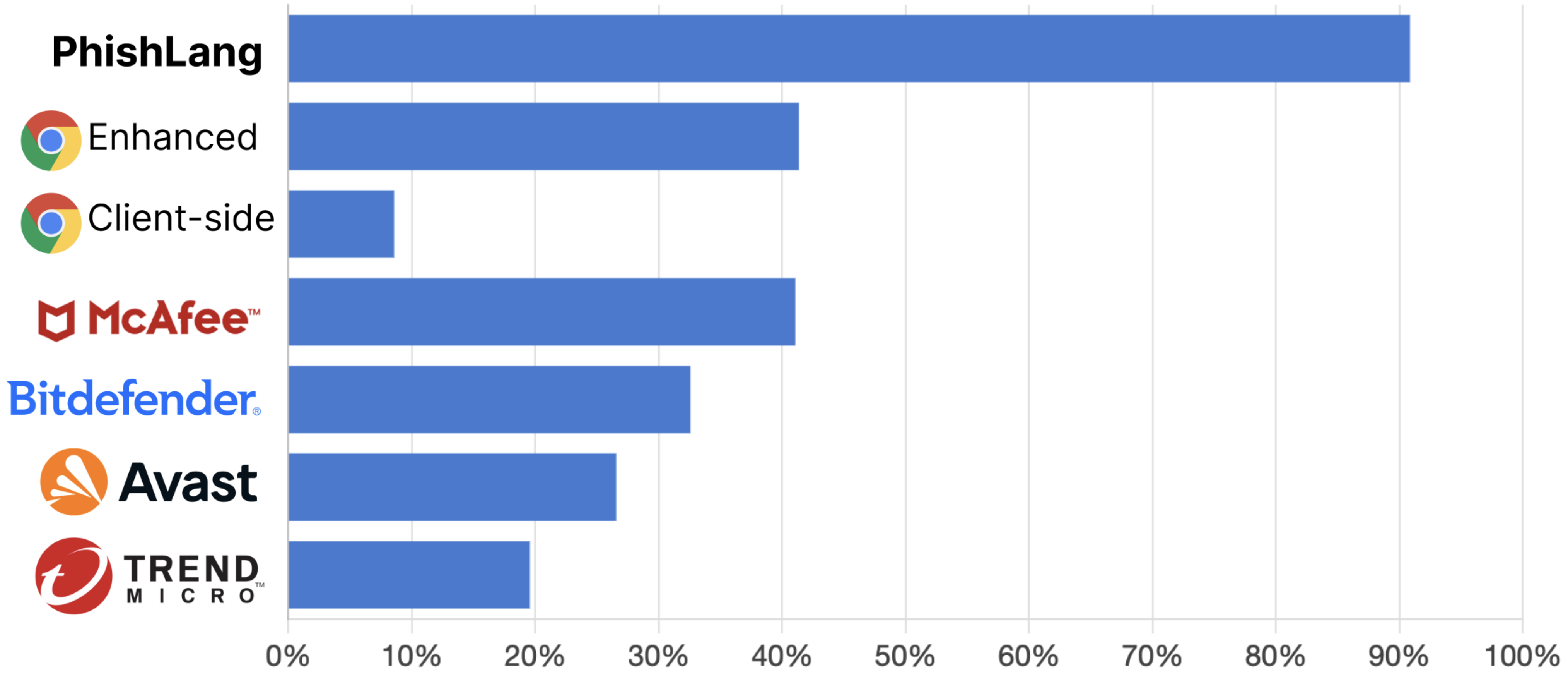
Bitdefender 35%

Avast 38.4%

TREND MICRO™ 26%

https://dl.acm.org/doi/pdf/10.1145/3634737.3657027

Against evasive threats



Conclusion

Introduced PhishLang, a language model based lightweight, fully client-side phishing detection framework.

Focuses on actionable HTML elements to model phishing intent beyond static heuristics.

Identified thousands of regular and evasive phishing URLs in real-time, including many missed by major blocklists

Demonstrates strong robustness against problem-space adversarial attacks

Runs efficiently on low-end consumer systems as a browser extension

- Outperforming several consumer-tools in zero-day detection.

Thank you!

sayak.saharoy@lsu.edu
shirin.nilizadeh@uta.edu

Download PhishLang
https://github.com/UTA-SPRLab/phishlang

Read the paper