Evading Anti-Phishing Models: A Field Note Documenting an Experience in the Machine Learning Security Evasion Competition 2022

YANG GAO, Indiana University, USA
BENJAMIN M. AMPEL, University of Arizona, USA
SAGAR SAMTANI, Indiana University, USA

Although machine learning-based anti-phishing detectors have provided promising results in phishing website detection, they remain vulnerable to evasion attacks. The Machine Learning Security Evasion Competition 2022 (MLSEC 2022) provides researchers and practitioners with the opportunity to deploy evasion attacks against anti-phishing machine learning models in real-world settings. In this field note, we share our experience participating in MLSEC 2022. We manipulated the source code of ten phishing HTML pages provided by the competition using obfuscation techniques to evade anti-phishing models. Our evasion attacks employing a benign overlap strategy achieved third place in the competition with 46 out of a potential 80 points. The results of our MLSEC 2022 performance can provide valuable insights for research seeking to robustify machine learning-based anti-phishing detectors.

CCS Concepts: • **Security and privacy** → **Phishing;**

Additional Key Words and Phrases: Evasion attacks, adversarial machine learning, anti-phishing detectors, MLSEC 2022

ACM Reference format:
Yang Gao, Benjamin M. Ampel, and Sagar Samtani. 2024. Evading Anti-Phishing Models: A Field Note Documenting an Experience in the Machine Learning Security Evasion Competition 2022. *Digit. Threat. Res. Pract.* 5, 1, Article 7 (March 2024), 8 pages.
https://doi.org/10.1145/3603507

1 INTRODUCTION

Machine learning (ML)-based detection systems for phishing websites (i.e., malicious portals that mimic legitimate websites) have seen great progress in recent years [4]. However, studies have shown that ML-based anti-phishing classifiers are vulnerable to adversarial evasion attacks [3, 13]. Evasion attacks occur when ML models are fed carefully perturbed input samples with the goal of misclassification [5]. Therefore, researchers must be alert to the threat of evasion attacks when building ML-based anti-phishing classifiers.

The **Machine Learning Security Evasion Competition (MLSEC)** is an annual competition to study how adversaries can deploy evasion attacks against ML detectors in real-world settings. MLSEC 2022 (August 12-September 23) was hosted by Zoltan Balazs (Head of the Vulnerability Research Lab at CUJO AI), Hyrum

Authors' addresses: Y. Gao and S. Samtani, Indiana University, 1275 E. 10th St, Bloomington, IN, 47405; emails: {gaoyang, ssamtani}@iu.edu; B. M. Ampel, University of Arizona, 1130 E. Helen St., Tucson, AZ, 85719; email: bampel@arizona.edu.

2576-5337/2024/03-ART7 $15.00
https://doi.org/10.1145/3603507

Anderson (Distinguished Engineer at Robust Intelligence), and Eugene Neelou (Co-Founder and CTO at Adversa AI) [11]. In the anti-phishing track of MLSEC 2022, researchers from CUJO AI provided eight HTML-based anti-phishing models and ten original phishing web pages (in HTML) to participants. The anti-phishing models were trained by interpreting HTML and extracting the **Document Object Model (DOM)** [7]. The goal of the competition was to evade the anti-phishing models by perturbing the web pages while maintaining the pages' rendering. In this field note, we report our experiences and the strategies we employed to earn third place in the competition.

The remainder of this field note is organized as follows: First, we describe related work employing evasion techniques to attack ML-based models. Second, we detail the strategies and steps we took in the MLSEC 2022 competition. Third, we discuss the insights derived from the competition that could help facilitate future research. Finally, we conclude the work and consider possible future directions.

2 RELATED WORK

Anti-phishing classifiers are often built with ML architectures and are trained by extracting features from URLs, the source code of web pages, or screenshots of web pages [1, 16]. ML models, despite their potential and versatility, have vulnerabilities that can be exploited by evasion attacks [2, 4, 5]. Attackers can carefully manipulate the features of phishing websites to cause ML models to misclassify phishing web pages as legitimate [3, 8, 13]. Prior studies have demonstrated that attackers employ code obfuscation techniques (e.g., encoding phishing content and injecting dead code) to conceal phishing content within HTML source code [9, 13, 17]. Top teams in MLSEC 2021 embedded encoded phishing content within benign content (e.g., Wikipedia pages) [6]. For example, the first-place and second-place teams encoded phishing content with Base64 and executed it with JavaScript. Base64 encoding is a frequently used obfuscation technique exploited by phishers to encode HTML content [15]. They then used JavaScript to remove benign content during web page rendering. This process prevented the benign content from being displayed while decoding and executing the phishing content. The JavaScript strategy successfully masked phishing content and allowed the manipulated HTML files to evade all anti-phishing models [6]. Therefore, we explored code obfuscation with JavaScript to discover if this strategy still worked in MLSEC 2022.

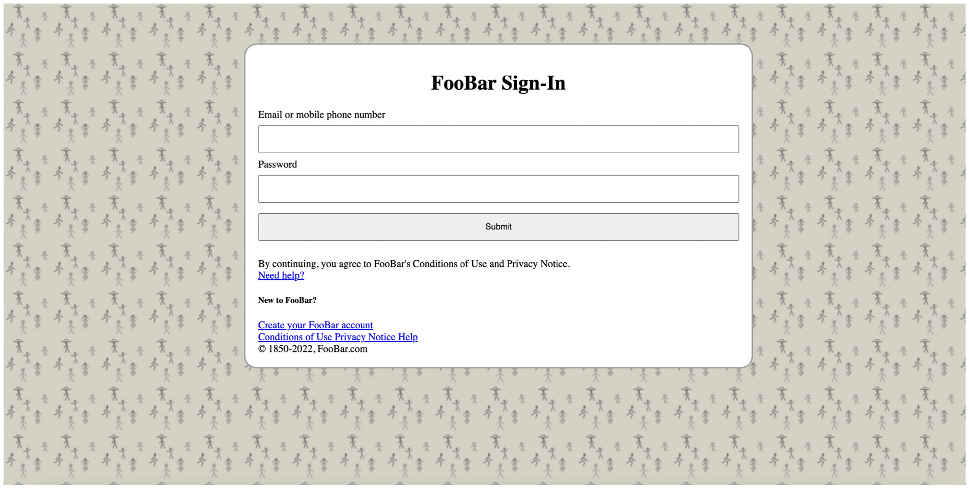
3 EVASION ATTACKS

3.1 MLSEC 2022 Competition Details and Preliminary Analysis

Each anti-phishing model in the MLSEC 2022 competition outputs a decision score between 0-1. Successful evasion of a web page requires an anti-phishing model to output a decision score below 0.1. Participants earn 1 point for each successful evasion. In total, participants could earn a total of 80 points (ten phishing HTML pages across eight anti-phishing models; full details are at: [11]). Each phishing web page includes a repeated background, links to external sources, plain text, and an input form. We show a rendering of a web page in Figure 1. Before introducing any manipulations, we tested the performance of the anti-phishing models on the original phishing pages. In particular, we fed the first phishing web page (Figure 1) into all anti-phishing models. The results are shown in Table 1. Overall, all models produced high decision scores (ranging from 0.744 to 0.926) on the first phishing web page.

3.2 Initial Evasion Attempts: Code Obfuscation with JavaScript

In the MLSEC 2022 competition, all anti-phishing models were trained using features derived from HTML source code. As a result, evasion techniques that manipulate HTML may cause the model to misclassify a phishing web page as benign. We chose the benign HTML content of the Zhejiang University (ZJU) Library website [14] to place into each web page's HTML source code. Our initial evasion attempt followed the same strategies as the MLSEC 2021 top performers to use Base64 encoding to obscure phishing content within benign content as well



FooBar Sign-In

Email or mobile phone number

Password

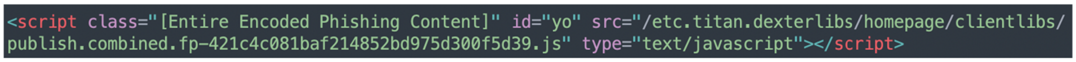
Submit

By continuing, you agree to FooBar's Conditions of Use and Privacy Notice.
Need help?

New to FooBar?

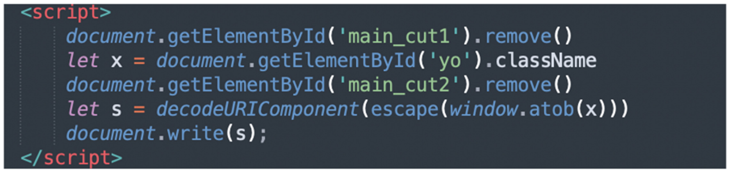
Create your FooBar account
Conditions of Use Privacy Notice Help
© 1850-2022, FooBar.com

Fig. 1. The rendering of web page 1.



```
<script class="[Entire Encoded Phishing Content]" id="yo" src="/etc.titan.dexterlibs/homepage/clientlibs/
publish.combined.fp-421c4c081baf214852bd975d300f5d39.js" type="text/javascript"></script>
```

Fig. 2. Entire encoded phishing content in <script>.



```
<script>
    document.getElementById('main_cut1').remove()
    let x = document.getElementById('yo').className
    document.getElementById('main_cut2').remove()
    let s = decodeURIComponent(escape(window.atob(x)))
    document.write(s);
</script>
```

Fig. 3. Example JavaScript code that decodes and executes phishing content.

Table 1. The Decision Scores of Web Page 1 Without Evasion Attempts

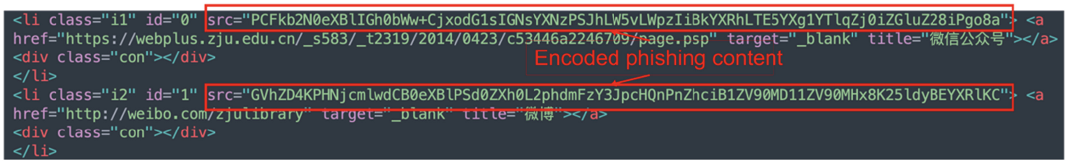
Web Page	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Web Page 1	0.926	0.885	0.926	0.885	0.766	0.744	0.766	0.744

as JavaScript to decode and execute the phishing content [6]. We encoded the HTML of each phishing web page with Base64 and inserted the encoded phishing content into the "class" attribute of a <script> element, shown in Figure 2.

We then implemented the JavaScript functions decodeURIComponent(), .remove(), and document.write() to remove benign HTML content from being displayed and decode phishing content for the browser to execute, shown in Figure 3.

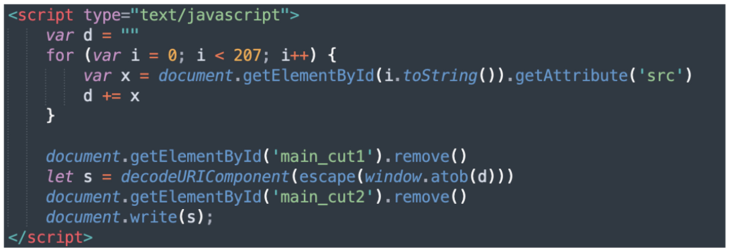
To further obscure the phishing content, we split the encoded content into fragments and inserted them into the "src" attributes in <li> elements of benign HTML, shown in Figure 4. We then called a JavaScript function to combine all fragments, which decoded the phishing content and removed the benign website, shown in Figure 5.

We implemented the manipulation strategy on all 10 phishing web pages and show the results of each model in Table 2. Overall, our manipulated phishing pages successfully evaded models 0 and 4 and almost evaded models 1



```
<li class="i1" id="0" src="PCFkb2N0eXBlIGh0bWw+CjxodG1sIGNsYXNzPSJhLW5vLWpzIiBkYXRhLTE5YXg1YTlqZj0iZGluZ28iPgo8a"> <a
href="https://webplus.zju.edu.cn/_s583/_t2319/2014/0423/c53446a2246709/page.psp" target="_blank" title="微信公众号"></a>
<div class="con"></div>
</li>
<li class="i2" id="1" src="GVhZD4KPHNjcmlwdCB0eXBlPSd0ZXh0L2phdmFzY3JpcHQnPnZhciB1ZV90MD11ZV90MHx8K25ldyBEYXRlKC"> <a
href="http://weibo.com/zjulibrary" target="_blank" title="微博"></a>
<div class="con"></div>
</li>
```

Fig. 4. Segmented encoded phishing content in <li>.



```
<script type="text/javascript">
    var d = ""
    for (var i = 0; i < 207; i++) {
        var x = document.getElementById(i.toString()).getAttribute('src')
        d += x
    }

    document.getElementById('main_cut1').remove()
    let s = decodeURIComponent(escape(window.atob(d)))
    document.getElementById('main_cut2').remove()
    document.write(s);
</script>
```

Fig. 5. JavaScript to combine segments, remove benign website, and execute phishing content.

Table 2. Decision Scores for Code Obfuscation with JavaScript

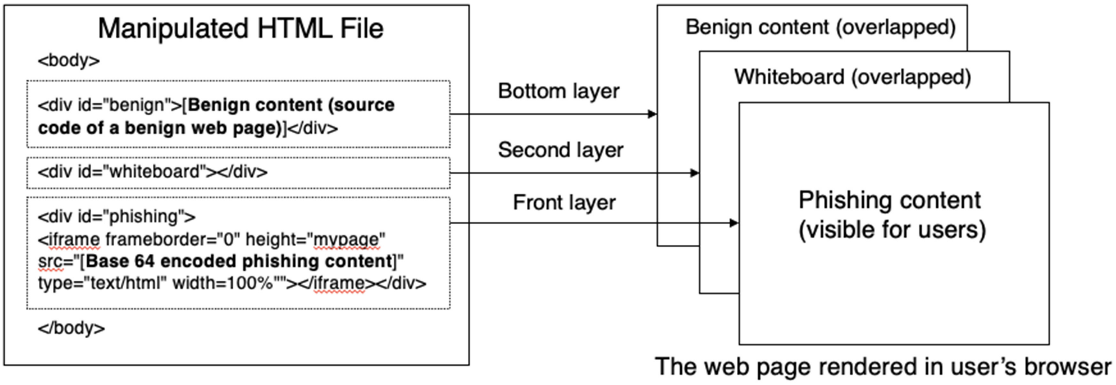
Web Page	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Web Page 1	**0.009**	0.177	0.924	0.930	**0.012**	0.199	0.845	0.795
Web Page 2	**0.009**	0.177	0.969	0.964	**0.012**	0.199	0.947	0.947
Web Page 3	**0.009**	0.182	0.434	0.713	**0.012**	0.204	0.698	0.747
Web Page 4	**0.009**	0.177	0.726	0.753	**0.012**	0.199	0.695	0.725
Web Page 5	**0.009**	0.177	0.748	0.857	**0.012**	0.199	0.715	0.835
Web Page 6	**0.009**	0.177	0.635	0.770	**0.012**	0.199	0.753	0.788
Web Page 7	**0.010**	0.210	0.969	0.985	**0.010**	0.201	0.937	0.964
Web Page 8	**0.010**	0.210	0.842	0.932	**0.010**	0.201	0.846	0.934
Web Page 9	**0.010**	0.210	0.967	0.983	**0.010**	0.201	0.937	0.962
Web Page 10	**0.010**	0.178	0.886	0.846	**0.012**	0.204	0.672	0.800

and 5. However, models 2, 3, 6, and 7 produced high decision scores. Overall, despite being largely successful in the MLSEC 2021 competition, the code obfuscation strategy using JavaScript did not achieve a great score (20/80).

The results of this initial exploration suggested that the anti-phishing models were retrained to be resilient against the evasion techniques used in the previous year's competition. In addition to training on static HTML features, the models in MLSEC 2022 also extracted DOM features, which may have increased their sensitivity to JavaScript. Therefore, we present an innovative benign overlap strategy that can conceal and render phishing content within benign content without using JavaScript.

3.3 Improved Attempt: Benign Overlap

In order to improve evasion performance, we developed a benign overlap strategy that does not rely on JavaScript to execute phishing content. Instead, both benign and phishing content is executed in the user's browser. However, benign content is visually overlapped by phishing content, such that users only see and interact with



```
<body>
<div id="benign">[Benign content (source
code of a benign web page)]</div>
<div id="whiteboard"></div>
<div id="phishing">
<iframe frameborder="0" height="mypage"
src="[Base 64 encoded phishing content]"
type="text/html" width=100%""></iframe></div>
</body>
```

Fig. 6. The framework of benign overlap strategy.

Table 3. Decision Scores for Benign Overlap (ZJU Library)

Web Page	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Web Page 1	**0.009**	0.188	**0.015**	0.348	**0.011**	0.242	**0.014**	0.369
Web Page 2	**0.009**	0.188	**0.015**	0.348	**0.011**	0.242	**0.014**	0.369
Web Page 3	**0.009**	0.192	**0.015**	0.355	**0.011**	0.248	**0.014**	0.377
Web Page 4	**0.009**	0.188	**0.015**	0.348	**0.011**	0.242	**0.014**	0.369
Web Page 5	**0.009**	0.188	**0.015**	0.348	**0.011**	0.242	**0.014**	0.369
Web Page 6	**0.009**	0.188	**0.015**	0.348	**0.011**	0.242	**0.014**	0.369
Web Page 7	**0.010**	0.222	**0.019**	0.401	**0.011**	0.244	**0.014**	0.372
Web Page 8	**0.010**	0.222	**0.019**	0.401	**0.011**	0.244	**0.014**	0.372
Web Page 9	**0.010**	0.222	**0.019**	0.401	**0.011**	0.244	**0.014**	0.372
Web Page 10	**0.009**	0.188	**0.016**	0.348	**0.011**	0.248	**0.014**	0.377

the phishing content. To conduct the benign overlap strategy, we overlapped benign source code from the ZJU Library web page with phishing content, shown in Figure 6.

Three layers of web content are necessary to conduct our benign overlap strategy. First, benign content was placed into a <div> element as the bottom layer. Second, a whiteboard layer overlaid the benign content to visually obscure it. Third, Base64 encoded phishing content was converted to a Data URI link format (data:text/html;Base64) and embedded into the "src" attribute of an <iframe> element as the top layer. The data URI link format enables the user's browser to execute the encoded content without relying on JavaScript. When the browser displayed the web page, the benign content was hidden behind phishing content.

We implemented the benign overlap strategy on each web page and tested them against each anti-phishing model. The results are displayed in Table 3. The benign overlap successfully evaded models 0, 2, 4, and 6. Although models 1, 3, 5, and 7 were not evaded, the decision scores were significantly reduced when compared to the results of the initial attempt relying on JavaScript. Given its success, we explored how to further improve benign overlap performance.

3.4 Sensitivity Analysis on Benign Web Page Templates

To further improve the evasion performance of the benign overlap strategy, we explored several types of benign content. Benign web pages were selected from the Top Website Ranking of Similarweb [12]. We systematically applied the content from each top website to web page 1 and tested each model's performance, shown in Table 4.

Table 4. Decision Scores for Selected Websites in Benign Overlap

Benign Website	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
ZJU Library	**0.006**	0.209	**0.010**	0.371	**0.018**	0.461	**0.012**	0.427
Mihoyo game site	**0.077**	0.272	**0.040**	0.130	0.100	0.322	**0.040**	0.130
1point3acres	**0.001**	0.206	**0.001**	0.123	**0.002**	0.173	**0.002**	0.200
BBC news	**0.003**	0.586	**0.006**	0.763	**0.035**	0.659	**0.062**	0.787
China news	**0.002**	0.442	**0.002**	0.442	**0.009**	0.813	**0.003**	0.666
Taobao	**0.016**	0.173	**0.019**	0.461	**0.026**	0.167	**0.023**	0.509
Reddit	**0.002**	0.299	**0.001**	0.130	**0.011**	0.709	**0.040**	0.130
Netflix	**0.014**	0.545	**0.009**	0.519	**0.022**	0.437	**0.013**	0.603
Zoom	**0.004**	0.602	**0.040**	0.130	**0.025**	0.910	**0.040**	0.130
Pinterest	**0.002**	**0.036**	**0.002**	**0.040**	**0.020**	**0.063**	**0.030**	**0.092**

Table 5. Decision Scores for Benign Overlap (Pinterest.com)

Web Page	Model 0	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	View Equivalent
Web Page 1	**0.002**	**0.037**	**0.002**	**0.040**	**0.019**	**0.084**	**0.019**	**0.085**	TRUE
Web Page 2	**0.002**	**0.037**	**0.002**	0.130	**0.019**	**0.084**	**0.040**	0.130	FALSE
Web Page 3	**0.002**	**0.037**	**0.002**	**0.040**	**0.019**	**0.084**	**0.019**	**0.085**	TRUE
Web Page 4	**0.002**	**0.037**	**0.002**	**0.040**	**0.019**	**0.084**	**0.019**	**0.085**	TRUE
Web Page 5	**0.002**	**0.037**	**0.002**	**0.040**	**0.019**	**0.084**	**0.019**	**0.085**	TRUE
Web Page 6	**0.002**	**0.037**	**0.002**	**0.040**	**0.019**	**0.084**	**0.019**	**0.085**	FALSE
Web Page 7	**0.002**	**0.044**	**0.040**	0.130	**0.025**	0.129	**0.040**	0.130	TRUE
Web Page 8	**0.002**	**0.044**	**0.040**	0.130	**0.025**	0.129	**0.040**	0.130	TRUE
Web Page 9	**0.002**	**0.044**	**0.040**	0.130	**0.025**	0.129	**0.040**	0.130	FALSE
Web Page 10	**0.002**	**0.047**	**0.040**	0.130	**0.021**	**0.094**	**0.040**	**0.096**	FALSE

From Table 4, we see that eight of the nine other benign web pages produce similar results as our original web page (ZJU Library). However, we found that using content from Pinterest.com [10] for benign overlap successfully evaded all eight models for web page 1.

Given the success of Pinterest.com in our prior test, we inserted its content into all ten web pages using the benign overlap evasion attack. All ten web pages were input into the eight anti-phishing models, and the results are shown in Table 5. As shown in Table 5, 67 decision scores dropped below 0.1. None of the decision scores was higher than 0.13, indicating that our model almost evaded detection across all web pages and models. Web pages 1, 6, 9, and 10 could not pass the view equivalent test, which is necessary to be awarded a point for evasion. However, the overall results suggest that this successful attack should be monitored by phishing detection experts.

4 OVERALL INSIGHTS

We visualized the results in Figure 7 to help provide an overall perspective of our attempts vs the benign overlap variations (ZJU Library and Pinterest.com). The green line indicates the evasion threshold. Our initial attempt replicated the code obfuscation relying on JavaScript used by the MLSEC 2021 winners. However, the updated anti-phishing models in MLSEC 2022 were robust against the JavaScript evasion attempts (blue bars in Figure 7). Our benign overlap strategy, which combined three layers of overlapping content with no JavaScript, led to a significant improvement over MLSEC 2021 strategies in evasion performance (orange bars in Figure 7). We then

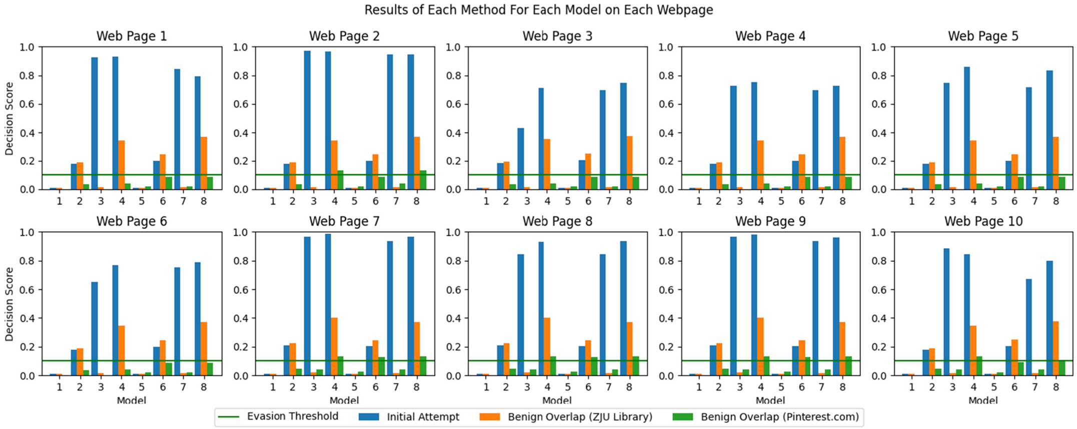


Fig. 7. Comparisons of initial attempt, benign overlap (ZJU library) and benign overlap (pinterest.com).

further refined the benign overlap strategy by using Pinterest.com as benign content, resulting in the evasion of more anti-phishing models.

Overall, our MLSEC 2022 Pinterest.com benign content strategy obtained 46/80 points, resulting in third place. Although the provided anti-phishing models were retrained on the basis of last year's submissions, statically trained models that extract features from source code can still be easily evaded with a few benign content strategies. Our benign overlap strategy encodes phishing content without relying on specific content, making it adaptable for a broader range of phishing pages. Benign overlap can be easily generalized to any number of phishing pages for extensive evasion campaigns. Therefore, there is an urgent need to measure the robustness of anti-phishing models by considering the adversarial manipulations presented in this Field Note.

5 CONCLUSION AND FUTURE DIRECTIONS

MLSEC has greatly benefited researchers by providing a platform to exercise evasion attacks in real-world settings. Our experience demonstrates that manual manipulations can achieve high evasion rates. By disseminating the results of these manipulations, researchers can build more robust anti-phishing models and further improve the competition.

Attackers are generating large numbers of phishing websites with multiple emerging evasion techniques. Even though anti-phishing models can be retrained based on manual manipulation experience, as was done based on MLSEC 2021 winners' results, only minor adjustments to the manipulation method can still evade the models, as demonstrated in our experience of MLSEC 2022. Therefore, manual manipulations based on human guesses may be inefficient in examining the vulnerabilities of anti-phishing models. In future research, researchers could develop automated adversarial learning models based on reinforcement learning or generative adversarial networks that can systematically apply various evasion techniques to generate large amounts of phishing websites to mimic attacker tactics. This research could greatly benefit the ongoing efforts to combat increasing phishing attacks.

ACKNOWLEDGMENTS

We would like to thank the hosts of MLSEC 2022 for providing a platform for researchers to systematically exercise phishing evasion attacks in practice. We also gratefully acknowledge Hyrum Anderson, Zoltan Balaza, and Eugene Neelou for the competition and their kind guidance throughout this process.

REFERENCES

[1] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. VisualPhishNet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS'20)*. ACM, New York, 1681–1698.

[2] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, and Mirco Marchetti. 2019. Addressing adversarial attacks against security systems based on machine learning. In *Proceedings of the 2019 11th International Conference on Cyber Conflict (CyCon'19)*, Vol. 900. ieeexplore.ieee.org, 1–18.

[3] Giovanni Apruzzese, Mauro Conti, and Ying Yuan. 2022. SpacePhish: The evasion-space of adversarial attacks against phishing website detectors using machine learning. *arXiv preprint arXiv:2210.13660* (2022).

[4] Giovanni Apruzzese, Pavel Laskov, Edgardo Montes de Oca, Wissam Mallouli, Luis Búrdalo Rapa, Athanasios Vasileios Grammatopoulos, and Fabio Di Franco. 2022. The role of machine learning in cybersecurity. *Digital Threats: Research and Practice* (Jul 2022). https://doi.org/10.1145/3545574 Just Accepted.

[5] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 387–402.

[6] CUJOAI. 2021. Announcing the Winners of the 2021 MLSEC. https://cujo.com/announcing-the-winners-of-the-2021-machine-learning-security-evasion-competition/.

[7] CUJOAI. 2022. MLSEC 2022-The Winners and Some Closing Comments. https://cujo.com/mlsec-2022-the-winners-and-some-closing-comments/.

[8] Bin Liang, Miaoqiang Su, Wei You, Wenchang Shi, and Gang Yang. 2016. Cracking classifiers for evasion: A case study on the Google's phishing pages filter. In *Proceedings of the 25th International Conference on World Wide Web (WWW'16)* (Montréal, Québec, Canada). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 345–356.

[9] Peng Peng, Limin Yang, Linhai Song, and Gang Wang. 2019. Opening the blackbox of VirusTotal: Analyzing online phishing scan engines. In *Proceedings of the Internet Measurement Conference* (IMC'19) (Amsterdam, Netherlands). ACM, New York, NY, USA, 478–485.

[10] Pinterest. 2022. Pinterest Homepage. https://www.pinterest.com/.

[11] RobustIntelligence. 2022. ML Security Evasion Competition 2022. https://www.robustintelligence.com/blog-posts/ml-security-evasion-competition-2022.

[12] Similarweb. 2022. Top Website Ranking. https://www.similarweb.com/top-websites/.

[13] Fu Song, Yusi Lei, Sen Chen, Lingling Fan, and Yang Liu. 2021. Advanced evasion attacks and mitigations on practical ML-based phishing website classifiers. *International Journal of Intelligent Systems* 36, 9 (2021), 5210–5240.

[14] Zhejiang University. 2020. Library Website of Zhejiang University. https://libweb.zju.edu.cn/.

[15] Andrea Venturi, Michele Colajanni, Marco Ramilli, and Giorgio Valenziano Santangelo. 2022. Classification of Web Phishing Kits for early detection by platform providers. *arXiv preprint arXiv:2210.08273* (2022).

[16] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. 2011. CANTINA+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Trans. Inf. Syst. Secur.* 14, 2 (Sept. 2011), 1–28.

[17] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, R. C. Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2021. CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing. In *Proceedings of the2021 IEEE Symposium on Security and Privacy (SP'21)*. ieeexplore.ieee.org, 1109–1124.

Received 1 December 2022; revised 13 May 2023; accepted 18 May 2023