

Proactive Detection of Impostor Domains for Improved Protection Against Spear Phishing Attacks

Jeremy Dean

Student - Department of Computer Science, Texas A&M University–Corpus Christi, Corpus Christi, USA
jeremy.dean@tamucc.edu

Abstract—Spear phishing attacks pose a persistent threat to organizations, with significant financial losses reported annually. Despite advancements in detection methods, spear phishing campaigns continue to thrive. This research addresses the need for proactive measures to identify impostor domains soon after they are registered, thus preventing spear phishing attacks. By introducing a novel approach using Python-based tools, DNS Twist and domainCloneChecker (DCC), this study evaluates the effectiveness of detecting malicious domains shortly after registration. The methodology involves simulating potential targets for small businesses across different sectors and assessing the tools' performance in identifying impostor domains. Comparisons with existing methodologies reveal a 10% minimum improvement in spear phishing domain detection. The simplicity and low-cost implementation of this approach make it particularly valuable for small businesses lacking enterprise-level email protection. The results suggest that proactively scanning for impostor domains can significantly enhance spear phishing protection and should be considered for further exploration and integration into existing cybersecurity strategies.

I. INTRODUCTION

Spear phishing attacks continue to plague organizations. Numerous advancements have been made over the years to detect and prevent spear phishing attempts, yet for 2022 the FBI reported over three hundred thousand phishing victims reported losses of over fifty-two million dollars [1]. These are only losses that companies that *reported* to the FBI, meaning the true impact of phishing attacks on businesses is even higher.

There are several ways to improve protection against such attacks on several fronts including employee and end-user training, advanced heuristics and machine learning techniques, and ever-improving public blacklists [2]. Default settings in most modern web browsers will alert a user if they navigate to a flagged domain. However, many spear phishing campaigns occur exclusively over email, and many times the campaign is over and done with long before a detection is made. How can a business know when they are being targeted? Or, conversely, when a phishing campaign is targeting their customers, partners, suppliers, or service providers while posing as them?

Take this example: The accounts payable clerk at Acme Brick receives an email from wile.e.coyote/@looneytunes.com with his usual email signature asking that she re-route the usual commission payment to a new ACH account. Will she notice that the email is NOT from his usual @looneytunes.com

domain? Especially if a previous email chain is pasted beneath the new message?

In this example, ACME Brick has been compromised, and the bad actor has obtained access to an email thread, then legally registered a similar-looking domain to that of a contractor, and is now pretending to be picking up on a previous email thread. Wile E. Coyote was not compromised in any way but will nonetheless be negatively impacted by this exchange if the scam is successful.

This example mirrors a real-world case that happened to a colleague in March 2023. Based on the FBI numbers quoted in the opening paragraph we can safely assume similar spear phishing campaigns are underway every day all over the world.

How valuable would it be if a business received an automated alert that could auto-blacklist domains that are visually similar to domains they own or domains they frequently interact with, soon after the domain is registered? Such a proactive approach would thwart spear phishing attacks before they start, and also alert IT and Cybersecurity staff of the incident, providing knowledge that could then be shared out to their business contacts as suitable, and perhaps sent up to several public blacklists, much the way phishing websites are reported by other solutions [2].

Most enterprise-level organizations likely perform similar detection as suits them, with various degrees of success, either in-house or through a third party. However, small businesses with small or non-existent IT departments are left to rely on whatever protections and detection their email service provider has put in place, which are not always effective [5]. Since an email from a legitimately registered domain is not in itself suspicious, it can be complicated to identify and detect spear phishing emails.

In this paper, we review the effectiveness of proactive impostor domain detection. We utilize two Python programs – an open-source Python-based tool called DNS Twist [6], and an in-house tool that is still under development known as DCC (short for DomainCloneChecker) [7]. Such tools can identify malicious look-a-like impostor domains shortly after they are registered. In doing so, the new domain can be blacklisted and/or reported before the first email is sent in. Such a methodology can be implemented at little to no cost by any organization to bolster spear phishing protection. We will go over these tools in Section II.

In this paper, we will go over tools that can find impos-

tor domains from a list of legitimate domains, gauge their effectiveness for three fabricated small businesses in different sectors, and review the results.

II. RELATED WORK

To prepare for this research, we looked over several scholarly articles about phishing detection. Advanced algorithms and frameworks for detection of phishing websites have been developed and proven successful [2] [3] [4] using an assortment of checks to both identify those sites as well as find ways some sites evade detection and blacklisting. While these tools do check for various properties consistent with impostor URLs, they are not proactive in their approach and are only checking for very generic properties (subdomain, numbers in URL, etc.) which are not thorough and, by design, do nothing for sites without A or AAA records.

Therefore, we put forth a novel method for developing a list of likely impostor domains based on mutations of those domains a company already communicates with, then checking to see if these are registered. Such a process could further bolster the effectiveness these other methods offer.

Specifically, the article “Phishing website detection using a novel multipurpose dataset and web technologies features” [2] uses URL analysis and considers the following properties: subdomain, IP address, TLD, length, and special chars. None of these features do anything for a domain that is legally registered as a one-off of a legitimate domain (recall loonytunes.com example in Section I). There is also nothing proactive about this approach – scanning the domains using such a tool as emails come in would involve additional work/automation/maintenance.

“PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists” [3] is focused on the effectiveness of blacklists. It uses a framework to deploy innocuous phishing sites, then tests the efficiency and detection of those sites by three primary blacklists: GSB, SmartScreen and Opera, on both desktop and mobile browsers. While these blacklists are browser-based and therefore not one-to-one with email filtering, it is safe to assume sites on this blacklist can be blocked through some method/service making use of these blacklists. However, it does nothing if the malicious sites are not on the blacklist, and the blacklists are developed by crawlers, meaning once again, domains without A/AAAA records will not be detected.

“Kn0w Thy Doma1n Name: Unbiased Phishing Detection Using Domain Name Based Features” [4] is most closely related to our research. However, this solution is again reactive; they are determining how well a trained model can predict if a given site is a phishing site, and use domain length and URL length along with other measures which are only relevant if a domain has a website. This does nothing for the detection of impostor domains which are only set up with the intent of cloaking a bad actor with a familiar-looking domain as they send carefully crafted emails to advance a spear phishing campaign.

None of these three solutions focus on email or spear phishing attacks, and therefore it’s not fair to say they “fall short” on spear phishing domain detection; however, these methodologies will not adequately identify impostor domains that only hold MX records and use simple mutations of valid domains.

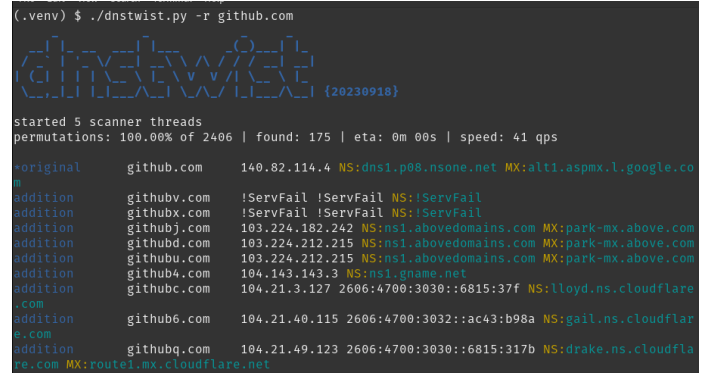
III. TOOLS AND METHODOLOGY

Our research involves the use of three core tools: DNS Twist [6], domainCloneChecker (DCC) [7] for impostor domain detection, and WHOIS JSON API [8] to pull registrar information.

A. DNS Twist

We discovered DNS Twist while working to improve our own Python scripts for impostor domain detection, and found that it did much of what we were attempting to do and more. It works by adding, removing and substituting characters to a given domain and checking/reporting is that combination exists. Since it is an open source solution, we decided to incorporate it into this project and use it as the first step for detection. The GitHub page sums up the scanners capabilities nicely, explaining it can “detect typosquatters, phishing attacks, fraud, and brand impersonation. Useful as an additional source of targeted threat intelligence.”

DNS Twist offers a variety of options and arguments, but for our research, we used the base option which scans the .com TLD only. Sample result in Figure 1.



```
(.venv) $ ./dnstwist.py -r github.com
started 5 scanner threads
permutations: 100.00% of 2406 | found: 175 | eta: 0m 00s | speed: 41 qps
+original      github.com      140.82.114.4 NS:ns1.p08.nsone.net MX:alt1.aspmx.l.google.co
+addition      githubv.com     !ServFail !ServFail NS:!ServFail
+addition      githubx.com     !ServFail !ServFail NS:!ServFail
+addition      githubj.com     103.224.182.242 NS:ns1.abovedomains.com MX:park-mx.above.com
+addition      githubd.com     103.224.212.215 NS:ns1.abovedomains.com MX:park-mx.above.com
+addition      githubu.com     103.224.212.215 NS:ns1.abovedomains.com MX:park-mx.above.com
+addition      github4.com     104.143.143.3 NS:ns1.gname.net
+addition      githubc.com     104.21.3.127 2606:4700:3030::6815:37f NS:lloyd.ns.cloudflare
+addition      github6.com     104.21.40.115 2606:4700:3032::ac43:b98a NS:gail.ns.cloudflare
+addition      githubq.com     104.21.49.123 2606:4700:3030::6815:317b NS:drake.ns.cloudflare
+addition      MX:route1.mx.cloudflare.net
```

Fig. 1. DNS Twist basic operation

B. domainCloneChecker

This is our in-house script. Similar to DNS Twist, it checks permutations of a given domain root or list of roots. It stands apart from DNS Twist in that it can be customized via a config.json file to do specific substitutions and TLDs. For our research, we used these substitutions and suffixes as seen in Figure 2. While there is considerable overlap in DCC results with DNS Twist, we do see new domains identified. While DNS Twist is a much more mature tool, we value the customization of DCC. Sample run shown in Figure 3.

```

"substitutions": {
  "1": ["1", "l"],
  "2": ["2", "z"],
  "3": ["3", "e"],
  "4": ["4", "a"],
  "5": ["5", "s", "z"],
  "6": ["6", "g"],
  "7": ["7", "t"],
  "8": ["8", "b"],
  "9": ["9", "p"],
  "0": ["0"],
  "a": ["4"],
  "b": ["8"],
  "c": ["c"],
  "e": ["3"],
  "g": ["6", "9"],
  "i": ["1", "l"],
  "l": ["1", "l"],
  "m": ["m"],
  "o": ["0"],
  "p": ["9"],
  "q": ["q"],
  "s": ["5", "z"],
  "t": ["7"],
  "v": ["v"],
  "z": ["2", "s"]
},
"suffices": [
  ".com",
  ".net",
  ".org",
  ".edu",
  ".io",
  ".tv",
  ".co",
  ".biz"
]

```

Fig. 2. DCC options in config.json

```

domainCloneChecker
2023-12-27 18:57:18,413 [INFO][run] Domain specified via command line: github.com
2023-12-27 18:57:18,413 [INFO][run] Root: github
2023-12-27 18:57:19,950 [INFO][run] Number of base strings to check: 7
2023-12-27 18:57:19,950 [INFO][run] Number of char substitutions coded: 1
2023-12-27 18:57:19,951 [INFO][run] Number of possibilities for char substitutions coded: 1
2023-12-27 18:57:20,372 [DEBUG][check_domain] github.com --found: A Record: 140.82.112.3 MX Record: 1 aspmx.l.google.com. 10 alt1.aspmx.l.google.com. 5 alt1.aspmx.l.google.com. 5 alt2.aspmx.l.google.com. SOA Record: dns1.p08.nsone.net. hostmaster.nsone.net. 1656468023 43200 7200 1209600 3600
2023-12-27 18:57:20,809 [DEBUG][check_domain] github.net --found: SOA Record: dns1.p05.nsone.net. hostmaster.nsone.net. 1648026590 43200 7200 1209600 3600
2023-12-27 18:57:22,221 [DEBUG][check_domain] github.org --found: A Record: 140.82.113.18 SOA Record: dns1.p09.nsone.net. hostmaster.nsone.net. 1646323670 43200 7200 1209600 3600
2023-12-27 18:57:22,221 [DEBUG][check_domain] github.io --not found: A Record: 185.199.109.153 185.199.110.153 185.199.111.153 185.199.108.153 SOA Record: dns1.p05.nsone.net. hostmaster.nsone.net. 1647625169 43200 7200 1209600 3600
2023-12-27 18:57:26,200 [DEBUG][check_domain] github.tv --found: A Record: 140.82.113.18 SOA Record: dns1.p06.nsone.net. hostmaster.nsone.net. 1646323672 43200 7200 1209600 3600
2023-12-27 18:57:26,605 [DEBUG][check_domain] github.co --found: A Record: 40.71.11.169 104.43.221.31 SOA Record: ns1-37.azure-dns.com. azure-dns-hostmaster.microsoft.com. 13600 300 2419200 300
2023-12-27 18:57:26,976 [DEBUG][check_domain] github.biz --found: A Record: 140.82.113.18 SOA Record: dns1.p03.nsone.net. hostmaster.nsone.net. 1646323667 43200 7200 1209600 3600
github.com --not found

```

Fig. 3. DCC basic operation

C. WHOIS JSON API

This tool is straightforward – it allows a script to query WHOIS data from its repository. This is important to establishing when an impostor domain was registered, as we likely are not as concerned with impostor domains that were registered long ago. Most Linux distributions come with whois built-in, however, a script to query whois information from hundreds or thousands of servers will quickly get blocked by rate limiting on the whois servers of various registrars. Therefore, it is important to use a service that allows us the effectively query the needed data without spamming whois servers across the world. Base usage is demonstrated in Figure 4.

D. Methodology

At first glance, the effectiveness of these tools may seem easy to evaluate, however, on further experimentation we determined it would be impossible to know if a tool tested for every possible mutation. DCC will only allow for up to

```

(.venv) $ ./checkwhois.py github.com
domain:
id: 1264983250_DOMAIN_COM-VRSN
domain: github.com
punycode: github.com
name: github.com
extension: com
whois_server: whois.markmonitor.com
status: ['clientdeleteprohibited', 'clienttransferprohibited', 'clientupdateprohibited']
name_servers: ['dns1.p08.nsone.net', 'dns2.p08.nsone.net', 'dns3.p08.nsone.net', 'dns4.p08.nsone.net', 'ns-1283.awsdns-32.org', 'ns-1707.awsdns-21.co.uk', 'ns-421.awsdns-52.com', 'ns-520.awsdns-01.net']
created_date: 2007-10-09T18:20:50Z
created_date_in_time: 2007-10-09T18:20:50Z
updated_date: 2022-09-07T09:10:44Z
updated_date_in_time: 2022-09-07T09:10:44Z
expiration_date: 2024-10-09T18:20:50Z
expiration_date_in_time: 2024-10-09T18:20:50Z
registrar:
id: 292
name: MarkMonitor Inc.
phone: +1.2086851750
email: abusecomplaints@markmonitor.com
referral url: http://www.markmonitor.com
(.venv) $

```

Fig. 4. Script using WHOISJSONAPI

two substitutions because going beyond that would take far too long to scan with the current algorithm.

To test the effectiveness of pro-active impostor domain detection, we decided to select a local small business, and then imagine what other businesses they would potentially feasibly do business with, then compile a list of those companies in a CSV with relevant information, including website URL.

For instance, we picked a local tire shop and guessed that they likely do business with several entities, including tire dealers, a cleaning service, a payroll service, state government, etc.

We then processed these CSV files through a series of programs to check and record impostor domain details. The results were used to build graphs and tables illustrating details about the impostor domains. Using this data, we determined whether impostors would have been detected by other means, as described in Section II. The process is detailed in Figure 5.

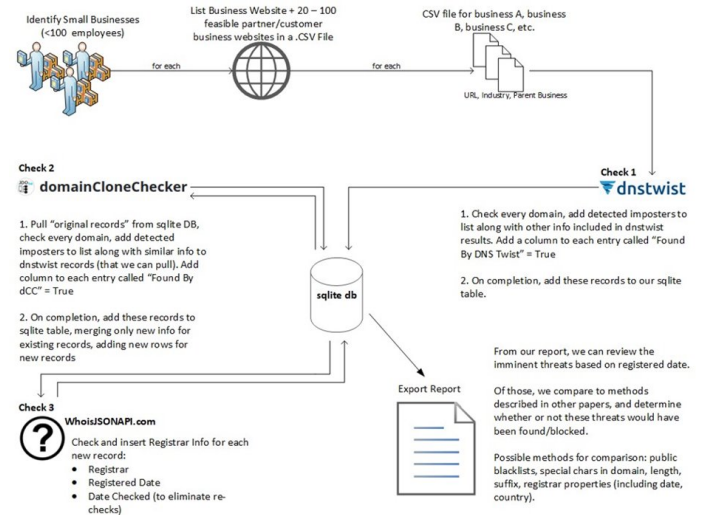


Fig. 5. Methodology for proactive detection of and reporting potential impostor domains

Note that neither DNS Twist nor DCC comes with op-

tions for SQLite integration. We built Python scripts to add/update/extract results from the database, and these scripts will be available along with instructions on the DCC GitHub page.

It's also worth noting that very short domain names (e.g., 3m.com) can produce a significant number of false positives. Fortunately, our focus is primarily on finding "active" spear phishing threats that are recently registered, allowing us to discard a good chunk of false positives. Very long domains can create thousands of mutations that must be scanned. Therefore, it may be wise to discard any domains longer than 20 characters from the initial list, depending on how disruptive a spear phishing campaign posing as that domain could be.

IV. RESULTS

Using the methodology outlined in Section III Part D, we were able to pull away the following metrics for our three test company datasets:

A. Company 1: Tire Company

We selected a local tire company and added 10 suppliers (tools + tires), 6 service providers (payroll, cleaning, tire recycling, etc) and 3 government sites (state inspection, employment, etc).

Results are in Tables I, II, III and Figure 6:

Company 1 - Tire Company	
Domains Scanned (including self)	20
Domains with Results	20
Total results (excludes original domains)	512
With Sites (A or AAAA)	457
With Email (MX)	197
With Site + Email	179
With Non-Alpha	52
With Non-Alphanumeric	27
Subdomain	57
Greater than 15 chars	110
Registered	
Last 2 years	134
Last 1 year	101
Last 3 months	18
Detected by DCC	113 97 exclusive
Detected by DNSTwist	415 399 exclusive

TABLE I
BREAKDOWN FOR COMPANY 1 - TIRE COMPANY

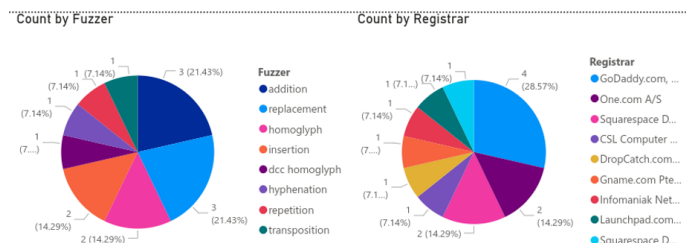


Fig. 6. Potential Spear Phishing Impostors by Fuzzer and by Registrar for Company 1 – Last 3 Months

How many of these are imminent for Spear Phishing Attacks?	
MX = True	12
DNS not like "park"	9
Standard suffix? (.com, .org, .net, .edu, .gov)	8
No numeric	7
No non-alphanumeric	7
No website	2
Not flagged on PhishTank	2

TABLE II
IMPOSTORS WHICH MAY BE SPEAR PHISHING SITES COMPANY 1

Summary:	
After surveying for 20 sites in this company profile, our solution accurately identified 18 Imposter domains targeting 9 companies scanned which were registered in the past 3 months.	
Of those, 0 were likely false positives,	
12 had MX records making them potential candidates for Spear Phishing senders	
Of those 12, 9 were not owned by parking domain companies	
Of those 9, 2 would not have been flagged by any researched methodology	
Therefore, our solution would have improved a Spear Phishing blacklist for an organization dealing with these 20 companies:	
Minimum Improvement in Detection	17% 2/12
(Assuming all other researched methodologies are implemented and detected all other imposters)	

TABLE III
SUMMARY AND MINIMUM IMPROVEMENT COMPANY 1

In Table 3, we see that Spear Phishing Site detection is improved at a minimum by 17% for the past three months. Keep in mind this number is very low to ensure we are being fair to the other methodologies. Without testing every impostor domain detected through every other phishing detection system, we can not determine if a site would actually have been detected. So we *assume* it would by default for our minimum improvement score. In reality, detection improvement would likely be much higher compared to whatever methods a particular company is (or is not) using to filter phishing emails today.

B. Company 2: Lumberyard

For our next test company, we selected a local lumberyard and added 3 suppliers (tools + lumber), 32 service providers (equipment, hvac, software, shipping, etc) and 15 customers (random big businesses).

Results are in Tables IV, V, VI and Figure 7:

Company 2 - Lumberyard	
Domains Scanned (including self)	50
Domains with Results	48
Total results (excludes original domains)	4855
With Sites (A or AAAA)	4470
With Email (MX)	1870
With Site + Email	1722
With Non-Alpha	877
With Non-Alphanumeric	296
Subdomain	185
Greater than 15 chars	665
Registered	
Last 2 years	887
Last 1 year	599
Last 3 months	116
Detected by DCC	572 423 exclusive
Detected by DNSTwist	4432 4283 exclusive

TABLE IV
BREAKDOWN FOR COMPANY 2 - LUMBERYARD

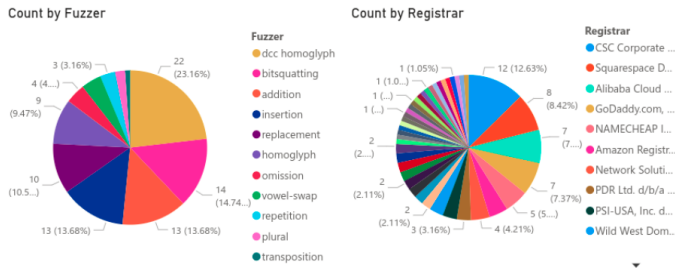


Fig. 7. Potential Spear Phishing Impostors by Fuzzer and by Registrar for Company 2 – Last 3 Months

Company 3 - ENT Clinic	
Domains Scanned (including self)	100
Domains with Results	96
Total results (excludes original domains)	1757
With Sites (A or AAAA)	1691
With Email (MX)	746
With Site + Email	709
With Non-Alpha	123
With Non-Alphanumeric	68
Subdomain	268
Greater than 15 chars	532
Registered	
Last 2 years	368
Last 1 year	263
Last 3 months	52
Detected by DCC	333
Detected by DNSTwist	1474
	283 exclusive
	1424 exclusive

TABLE VII
BREAKDOWN FOR COMPANY 3 - ENT CLINIC

How many of these are imminent for Spear Phishing Attacks?			
MX = True	72	can send legitimate emails	
DNS not like "park" (dns_mx and dns_ns)	64	parked domains typically not imminent threat	manual filter
Standard suffix? (.com, .org, .net, .edu, .gov)	59	many orgs bounce new TLDs	manual filter
No numeric	49	Know Thy DomaIn Name paper reviewed domain makeup deeming those with digits more likely to be phishing domain	
No non-alphanumeric	49	" " " non-alphanumeric " " "	
No website	6	PhishTime looked at website qualities so we just assume they simply would not have detected any site with NO website.	
Not flagged on PhishTank	5		fedexes.org flagged

TABLE V
IMPOSTORS WHICH MAY BE SPEAR PHISHING SITES COMPANY 2

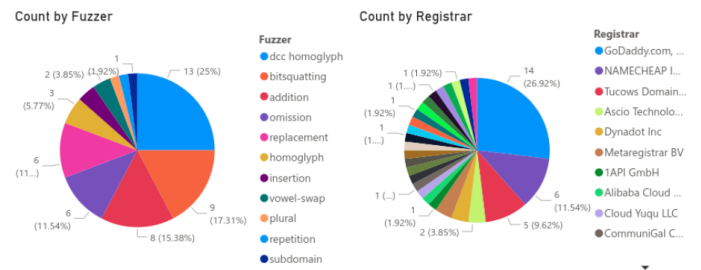


Fig. 8. Potential Spear Phishing Impostors by Fuzzer and by Registrar for Company 3 – Last 3 Months

Summary:	
After surveying 50 sites in this company's profile, our solution accurately identified 116 impostor domains targetting 30 companies scanned which were registered in the past 3 months.	
Of those, 0 were likely false positives,	
72 had MX records making them potential candidates for Spear Phishing senders	
Of those 72, 64 were not owned by parking domain companies	
Of those 64, 6 could not have been flagged by any researched methodology	
Therefore, our solution would have improved a Spear Phishing blacklist for an organization dealing with these 30 companies:	
Minimum	9% 6/64 (Assuming all other researched methodologies are implemented and detected all other imposters)

TABLE VI
SUMMARY AND MINIMUM IMPROVEMENT COMPANY 2

How many of these are imminent for Spear Phishing Attacks?			
MX = True	27	can send legitimate emails	
DNS not like "park" (dns_mx and dns_ns)	24	parked domains typically not imminent threat	manual filter
Standard suffix? (.com, .org, .net, .edu, .gov)	24	many orgs bounce new TLDs	manual filter
No numeric	19	Know Thy DomaIn Name paper reviewed domain makeup deeming those with digits more likely to be phishing domain	
No non-alphanumeric	19	" " " non-alphanumeric " " "	
No website	2	PhishTime looked at website qualities so we just assume they simply would not have detected any site with NO website.	
Not flagged on PhishTank			fedexes.org flagged

TABLE VIII
IMPOSTORS WHICH MAY BE SPEAR PHISHING SITES COMPANY 3

Summary:	
After surveying 100 sites in this company's profile, our solution accurately identified 96 impostor domains targetting 28 companies scanned which were registered in the past 3 months.	
Of those, 0 were likely false positives,	
27 had MX records making them potential candidates for Spear Phishing senders	
Of those 27, 24 were not owned by parking domain companies	
Of those 24, 2 could not have been flagged by any researched methodology	
Therefore, our solution would have improved a Spear Phishing blacklist for an organization dealing with these 28 companies:	
Minimum	8% 2/24 (Assuming all other researched methodologies are implemented and worked)

TABLE IX
SUMMARY AND MINIMUM IMPROVEMENT COMPANY 3

C. Company 3: ENT Clinic

For our last test company, we selected a local ENT clinic and added 11 suppliers (medical equipment + pharmaceuticals), 44 service providers (laboratories, other medical specialties, occupational health, IT services, etc), 6 government contacts and 33 customers (clinics and offices which refer patients).

Results are in Tables VII, VIII, IX and Figure 8:

D. Overall

Totaling the number of Potential Spear Phishing Domains found registered in the past 3 months, and looking at those that

could not have been identified by other researched methodologies, we can see in Tables X that 10% of these could not have been identified by other researched methods.

Again, this number is very low to ensure we are being fair to the other methodologies; we assume the other 90% would have been discovered for our minimum improvement score.

	Company 1	Company 2	Company 3		
TOTAL	2/12	6/64	2/24	=	10/100
10%	of the potential spear phishing impostor domains identified could not have been detected through other researched methods.				

TABLE X
MINIMUM IMPROVEMENT ALL 170 SITES SCANNED (FOR DOMAINS REGISTERED IN PAST 3 MONTHS)

V. CONCLUSION

From the research and tests performed, we conclude that performing proactive impostor domain scans can improve protection against spear phishing by 10% at a minimum. Many companies, especially small businesses, which this research is most interested in, likely have NO other phishing detection for their email besides what is configured by default by their service provider, which means this solution, by itself, could offer spear phishing detection where NO protection currently exists. Of 170 sites scanned, we found 186 potential impostor domains registered in the last 3 months, of those 100 had MX records; 58.8% of the sites scanned had an imposter domain that can send email, and therefore is potentially being used for spear phishing attempts against the impersonated domain.

We concede that 170 sites is not a large sample set and that further testing may find less favorable results. However, 10% minimum is a significant improvement and warrants further exploration at least.

Such an implementation could be easily automated by even novice IT Staff or an IT contractor with little to no cost for upkeep. A list of domains can be dumped from whatever email service is in place and then fed into DNS Twist and DCC to produce a list of impostor domains, which can then be fed into an email blacklist. The author of DNS Twist, Marcin Ulikowski, has confirmed with us by email that he is aware of several organizations that use his program for this purpose.

VI. FUTURE WORK

A. Further updates on DCC

domainCloneChecker was written by students piecemeal as this project has progressed since March 2023. As such, it needs to be re-worked and optimized to be multi-threaded, and functions added to incorporate several of the options we've written separate scripts for to export/import to/from CSV and SQLite.

We are also considering retiring the project and focusing on using DNS Twist exclusively, working with Marcin and the DNS Twist team to bring an any missing features.

B. Larger Sample Set

Once optimized, we should easily be able to run our algorithm on thousands of domains to produce more accurate estimations as to the effectiveness using a "top 1000 websites" list or similar input. This would help to support the case for this sort of pro-active detection.

C. Auto-Reporting to Phishing Blacklists

We've seen other frameworks report phishing sites on detection. We would like to incorporate an auto-report option which could be used by organizations which use this methodology to report impostor domains to public blacklists, such as PhishTank.org [9]

D. Blacklist Integration for O365 and Google Workspace

Similarly, it would be advantageous to build documentation for automating blacklist updates for detected impostor domains for major email service providers, especially Google and Microsoft. Such integration would allow small business owners to harness the power of these tools without technical knowledge or assistance.

REFERENCES

- [1] Federal Bureau of Investigation. Internet Crime Complaint Center (IC3) 2022 Annual Report. 2022. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2022_IC3Report.pdf
- [2] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, R. Alaiz-Rodríguez, "Phishing websites detection using a novel multipurpose dataset and web technologies features," *Expert Systems with Applications*, vol. 207, pp. 7-9, 2022.
- [3] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, A. Doupe, G.-J. Ahn, "PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists," in *29th USENIX Security Symposium*, 2020.
- [4] H. Shirazi, B. Bezawada, I. Ray, "Kn0w Thy DomaIn Name: Unbiased Phishing Detection Using Domain Name Based Features," in *Proceedings of the SACMAT '18: Proceedings of the 23rd ACM on Symposium on Access Control Models and Technologies*, pp. 69-73, 2018.
- [5] Barracuda Networks, Inc., "2023 Spear-Phishing Trends: Key findings about the impact of attacks and the challenges of threat detection and response," 2023. [Online]. Available: <https://assets.barracuda.com/assets/docs/dms/2023-spear-phishing-trends.pdf>
- [6] DNS Twist, GitHub Repository, 2022. [Online]. Available: <https://github.com/elceef/dnstwist>
- [7] domainCloneChecker, GitHub Repository, 2023. [Online]. Available: <https://github.com/deanj20/domainCloneChecker>
- [8] WhoisJSON API, "WhoisJSON API," n.d. [Online]. Available: <https://whoisjsonapi.com/>
- [9] PhishTank, "PhishTank," n.d. [Online]. Available: <https://phishtank.org/>