AFRL-RH-WP-TR-2023-0020

PREVENTING INFORMATION REMOVAL AND NABBING HARMFUL ACTORS (PIRANHA)

Marjorie Freedman / Genevieve Bartless
University of Southern California
Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90929

APRIL 2023

FINAL REPORT

AIR FORCE RESEARCH LABORATORY
711 HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AFB, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

Qualified requestors may obtain copies of this report from the Defense Technical Information Center (DTIC).

AFRL-RH-WP-TR-2023-0020 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

ANDERSON.TIMOTHY.RAY.1230210728

Digitally signed by ANDERSON.TIMOTHY.RAY.1230210728
Date: 2023.06.12 09:27:15 -04'00'

TIMOTHY R. ANDERSON, Ph.D., DR-IV
Work Unit Manager
Collaborative Technologies Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

STANARD.TERRY.W.1276780480

Digitally signed by STANARD.TERRY.W.1276780480
Date: 2023.06.12 09:37:53 -04'00'

TERRY STANARD, Ph.D., DR-III
Chief, Collaborative Technologies Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

CARTER.LOUISE.ANN.1230249128

Digitally signed by CARTER.LOUISE.ANN.1230249128
Date: 2023.06.21 07:57:40 -04'00'

LOUISE A. CARTER, Ph.D., DR-IV
Chief, Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
April 2023	Final	21 September 2018	28 February 2023

3. TITLE AND SUBTITLE
Preventing Information Removal and Nabbing Harmful Actors (PIRANHA)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8650-18-C-7878		
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		H0X4

6. AUTHOR(S)
Marjorie Freedman / Genevieve Bartless

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California
Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
711th Human Performance Wing
Airman Systems Directorate
Warfighter Interactions and Readiness Division
Wright-Patterson Air Force Base, OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'SREPORT NUMBER(S)
AFRL-RH-WP-TR-2023-0020

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION A: Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
AFRL-2023-2733; Cleared 5 Jun 2023

14. ABSTRACT
This report describes efforts under DARPA's ASED program in the detection of social engineering attacks. We describe the signals used for detection, experiments with those signals, and the results from the programs shared testbed. The detection approached developed during the program fused weak signals from the metadata, linguistic content of an email, and a grounding of who a sender is in the external world. The report also describes the engineering integration efforts that were used to integrate into the program's shared testbed. In the program testbed, the system deployed under this effort achieved 99% performance on the program's testbed.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	36

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Timothy Anderson, Ph.D.	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

This final report details the University of Southern California (USC) Information Sciences Institute (ISI) Preventing Information Removal and Nabbing Harmful Actors (PIRANHA) team's work on Defense Advanced Research Projects Agency (DARPA) Active Social Engineering Defense Program (ASED). The PIRANHA team worked on methods to detect phishing attacks across multiple channels and engage in dialogue to waste the scammer's time and extract information with the goal of using this information to link phishing campaigns.

2.0 INTRODUCTION

The ASED program worked towards identifying phishing attacks across multiple channels (email, short message service [SMS], and social media [SM]) and engaging in scammers through dialogue to waste the scammer's time and extract information with the goal of using this information to link phishing campaigns. Social Engineering (SE) threats, such as phishing, pose a massive security threat to government and industry alike. An estimated 80+% of successful data breaches in 2022 and 90+% of attacks on the banking industry starting with a social engineering attack (Ward & Subramnian, 2022).

The PIRANHA team was part of ASED's Technical Area 1 (TA1) (friend or foe detection) and TA2 (dialogue response) efforts with the majority of the team's effort on TA1. In Phase 1 of the program, all teams participated in both TA1 and TA2. In Phase 2, the PIRANHA team was the only team on the program working on TA1, with a continuing, but reduced effort on TA2.

PIRANHA's detection approach is based on finding weak signals which individually may not be enough to make a determination on friend or foe, but when combined can give confidence in the decision. In dialogue and response, the PIRANHA team focused on methods to augment neural dialogue approaches worked on by other teams on the ASED program. An overarching theme in PIRANHA's detection is combining meta-data cues with message content language cues to identify suspicious hints.

In addition to our research endeavors, our team also served as infrastructure support for the program in Phase 2. In Phase 1 of the ASED program, each performer team had their own unique architecture and deployment approach, with no common platform to standardize across. Furthermore, each team depended on a single integrator (Data Machines) to perform their deployment operations. This led to very disjointed and often chaotic evaluation periods, with a lot of wasted energy in operations not contributing to the research goals of the program.

As we entered Phase 2, the program adopted the goal of maintaining a persistent and continuously updated deployment presence in the evaluation testbed environment. To this end, the ISI/Northrup Grumman Corporation (NGC) team took the lead role in designing, building, and maintaining a continuously integrated (CI) arrangement of the ASED products, along with a suite of continuous deployment (CD) pipelines providing automated staging, test and promotion into the evaluation environment.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

In Sections 3.1 through. 3.4, we discuss four key signals the PIRANHA system utilized, the intuition behind these signals being useful in identifying phishing and how these signals are obtained.

- Firstly, the system gathers information and performs external vetting on this information to identify potential scams. This approach takes clues from the content of a message and attempts to place these clues within a broader context from information obtained from search engines and social media sites.
- Secondly, the system examines any uniform resource locators (URL) included in a message and explores linked websites to determine if the site is legitimate or a fly-by-night operation.
- We examine the style of a message and use stylometry to link known phishing messages to incoming messages based on similar style and also use stylometry to detect hijacked accounts.
- Lastly, the system promotes agenda pushing in automated responses to suspicious messages as a means of gathering more information to feed back into external vetting.

Section 3.5 discusses methods for putting weak signals together to make a binary friend or foe decision for a cohesive detection system.

In Section 3.6 we describe different approaches we took to analyzing linguistic elements of phishing attacks over the course of the program. Finally, in Section 3.7 we describe the engineering challenges and approach the PIRANHA team took to help integrate technologies with the program testbed and to integrate with transition partners.

3.1 External Vetting

Our approach to external vetting is to extract information from meta-data (largely email headers) and compare this information with content in the email and with online information.

In the simplest type of external vetting, we use the **From:**, **Return-Path:**, and **Reply-To:** fields in email headers and evaluate if the information in these fields makes sense for the labels given and if any of the domains given are associated with hijacked accounts or newly bought domains. Short-lived domains and email addresses with little history indicate a possible attack.

While each of these meta-data fields can be modified to hide the original origins of an email, this is most easily done for the **From:** field which most mail clients use to display who an email is purported to be from. Often attackers make the text of this field appear to be from an existing and legitimate organization (e.g., “Bank of America”). Adjusting the **Return-Path:**, and **Reply-To:** fields is not common in scams as at least one of these addresses needs to be under control of the attacker in order to receive replies.

We extracted the labels, email addresses, and the network domains found in these fields and used external databases collected from crawling SM accounts and data breach tracking sites to determine if the emails have a long history and are used on SM sites or other legitimate websites. We also checked to see if the domains were recently auctioned off or if they were new. Lastly,

we checked the labels for any legitimate organization names and used search engines to determine if the email address network domains match that of a domain owned by the legitimate organization. Mismatches between legitimate organization names and domains being used indicate an attack.

In addition to header fields, we found other clues that a message was being presented as being from a particular organization. The use of organization logos and/or an organization name in signatures can also identify when a message is supposed to present as being from a specific organization. To find and match logos used to organization we employed the Clearbit logo database and applied edge detection, logo rotation, and aspect skewing techniques to find and compare logos. This allows for matching logos that may have slight modifications or are placed in the background of pictures.

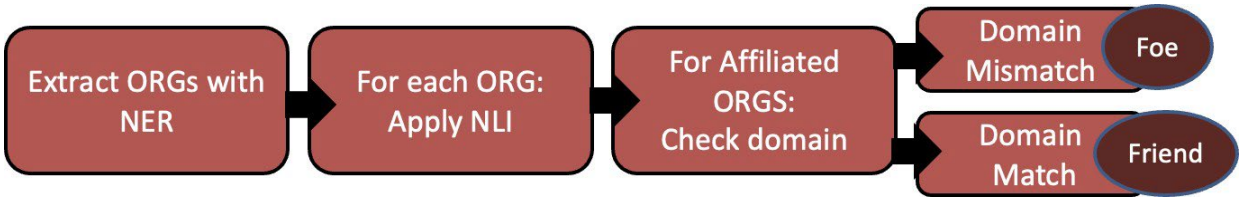

Figure 1: Pipeline for Extracting "EmployerOf(ORG, PER)" Relationships and Using this as a Signal for Friend/Foe.

Last, we explored using language cues in a message to identify which (if any) organization was being represented by the message in cases where the sender was claiming to work on behalf of the organization. To accomplish this, we used relation extraction to identify explicit **"EmployerOf(ORG, PER)"** relationships, as depicted in Figure 1. We did not use off-the-shelf relation extractors that are trained using the Automated Commercial Environment (ACE) dataset since these may not always be effective due to domain shifts and the implicit requirement of pronoun coreference. Instead, we used natural language inference (NLI) as a means of relation extraction. To accomplish this, context/hypothesis pairs are formed to check if an organization employs the email sender. We describe the technical approach to extracting EmployerOf() and other relations in more detail in Section 3.6.2.

For any of the above organization extractions (logo, signature, NLI), once we know a message claims to represent an organization or an individual speaking on behalf of an organization, we then can ensure other meta-data appears legitimate for that organization in the message. We used Bing business/maps application programming interfaces (API) to vet addresses and phone numbers included in such messages to be sure these matched the extracted organization. We identified the appropriate domain name system (DNS) domain based on the search engine ranking of the organization. The email sending domain and any links that the user is asked to click on should match this domain.

3.2 URL and Website Vetting

Understanding linked content in a message is a common need, as many phishing messages include *just* a link with little to no other content to evaluate. Similarly, phishing within specific applications insert web pages into app dialogue, leaving the web page itself as the only set of clues a scam is in place. While there is previous work in using website content to evaluate if

linked content is a scam, our work incorporates new features which are costly for the attacker to change to avoid detection.

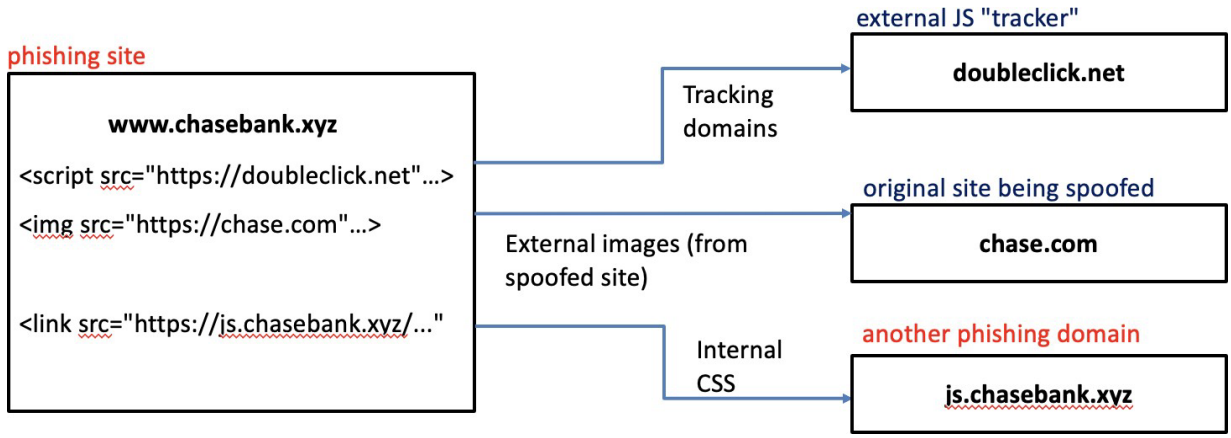

Figure 2: Example of a Spoofed Site and the Content it Pulls from External Sources.

Our work's key insight is that phishing sites often pull some content from the original site they are trying to spoof. Figure 2 shows an example of the content hosted by a spoofed site (*.chasebank.xyz) on attacker controlled domains vs the content the links of spoof site on original sites ("doubleclick.net" and "chase.com"). This copying of original content is done for several reasons, such as ease of deployment, automatically staying up to date with styles and logos, adding legitimacy with links to clearly legit content, and avoiding detection by potentially forwarding login to the real site after stealing credentials.

We trained a classifier to detect phishing sites based on featurized website structure. Our features emphasize the spread of the website across multiple domains, including intra-domains (domains under the same administrative control as the landing page) and inter-domains (domains external to the main site), as well as common hosting and tracking domains. We also used Secure Sockets Layer certificates (SSL certs) to assist in identifying what parts of a site are internal or external. To avoid detection, attackers would have to recreate entire portions of the original site, rather than just link to them, creating challenges in keeping spoofed sites up-to-date and performing similarly to the original site.

3.3 Stylometry and Phishing

Stylometry is the study of linguistic style—typically in written language. For our purposes, we focused on authorship, and employed stylometry in two ways. First, we worked to link phishing campaigns via stylometry authorship. Second, we demonstrated an efficient way to use stylometry to identify hijacked accounts. In phishing attacks, a hijacked email account is used to send out an attack. Typically, these attack messages are short and not stylized to match how the legitimate owner of the account writes messages.

We worked with two current featurization methods for stylometry: features based on tokenizing on a learned sub-word vocabulary and features based on pre-set tokenization using pre-trained word-embeddings. The learned sub-word vocabulary is language independent but did not perform as accurately in detecting hijacked accounts in testing, while the pre-set tokenization method is more accurate in detecting hijacked accounts and can produce explainable results

which highlight phrases and sentences that were drawn upon to identify two messages as coming from the same/different authors.

Our approach needed to be efficient in order to be a good candidate for evaluating incoming and outgoing emails at an email server. Our approach used distance learning to learn an embedded vector space. Authorship then was a matter of vectorization and vector comparisons—a task which has been highly optimized through multiple existing tool chains. We used NGT (a library for high-speed nearest neighbor searches over large vectors) (Yahoo! Japan, 2020) and were able to check 100k messages/day for hijacked accounts (outgoing email compared to an author's sent mail history) and for messages which stylistically matched known phishing attacks (1:many comparisons for each incoming message) without graphics processing units (GPU).

3.4 Agenda Pushing in Automated Dialogue

Part of the ASED effort was methods to engage attackers with automated dialogue to waste the attacker's time. We saw a need to have highly controlled *agendas* that could be inserted into conversations at appropriate times. Rather than fine-tune neural dialogue systems to handle specific scenarios, we wrote finite-state machines to orchestrate pushing agendas. We combined these transducers with neural dialogue which handles the open nature of conversations without needing to perform specific actions at the appropriate time.

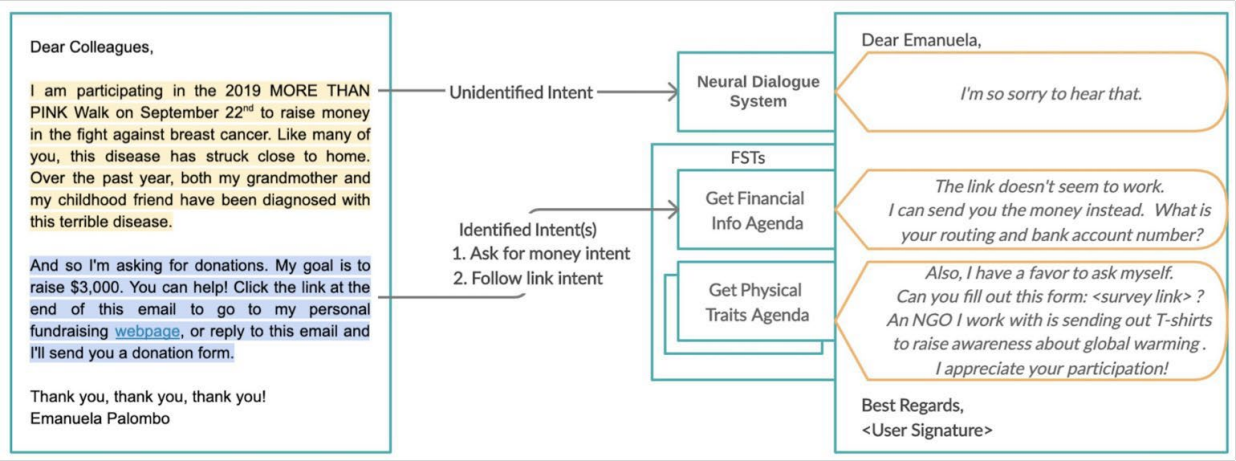

Figure 3: Example Combined Dialogue using Neural Dialogue with Agenda Pushing with Finite State Transducers (FST).

For our work, we utilized probabilistic FST where each state is associated with certain actions that are triggered when the probability of arriving at that state exceeds a configurable threshold. Our approach is capable of handling multiple agendas simultaneously. Figure 3 shows an example of an agenda. Figure 4 shows an example of the combined response between a neural dialogue engine and the transducer. A detailed look at our system can be found in our 2021 DialDoc publication (Cho et al., 2021).

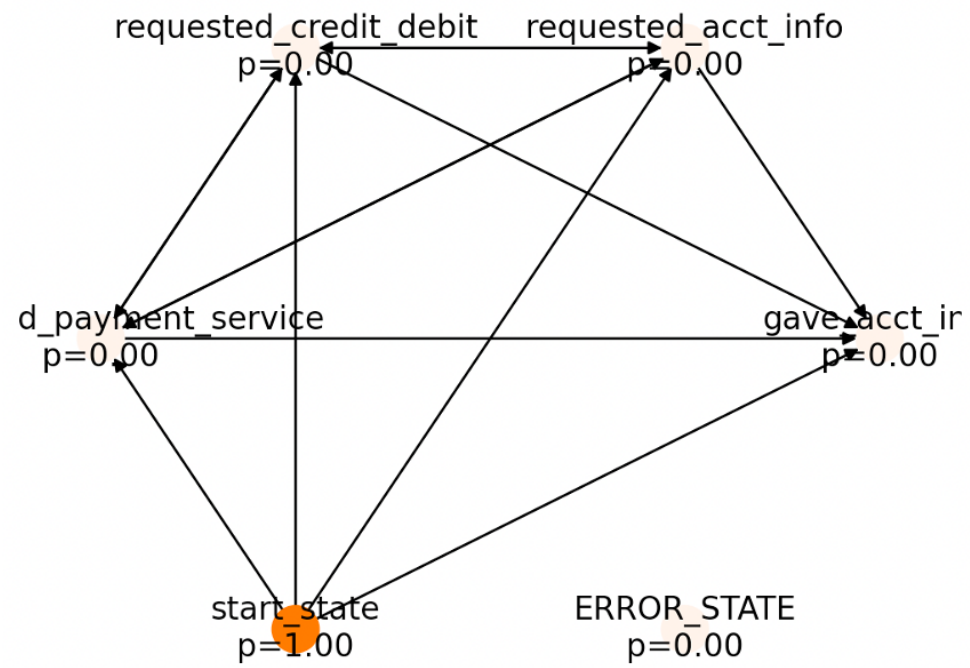

Figure 4: State Machine for "Get Payment Information" Agenda.

Our approach is informed by efforts by our UC Davis subcontractor team who worked with us in Phase 1 of the ASED program. Their efforts collected human-to-human dialogue using volunteers playing roles as scammer and scambaiters. These collected dialogues were then used to fine-tune Generative Pre-trained Transformer (GPT) to produce a neural dialogue engine which would respond appropriately to an Amazon helpdesk scam (Li et al., 2020). While the system responds fluently to the specific scam scenario, the dialogue can go off course in other scam scenarios. For example, when used in response to a scam about gift cards from Apple Music, the system will mention Amazon out of context. This going off subject lead us to using transducers to insert agendas. Our approach looks to be precise and seek information at appropriate points in dialogue through controlled statements and inserted based on when our agenda state machine indicates its needed, but otherwise carry a conversation through neural dialogue generation.

While we could apply fine-tuning to neural dialogue for a variety of situations, this would require significant data collection for each scenario. Instead, we create small agendas which are inserted at the appropriate times. Some agendas are applicable to many situations, such as an agenda to request a website link. This request applies to many online scams, such as an ask for a charity donation, a tax scam seeking to obtain private information, a fake lottery scam etc. Some agendas are more specific, such as asking the attacker for a shipping address, but most of these agendas can be used in multiple scenarios.

To ease the creation of new agendas to be inserted into conversation, our team created a graphical user interface (GUI) in which a user can drag and drop states and create an FST which can be loaded into our hybrid dialogue system.

3.5 Putting Signals Together to Make a Decision

Graph representation and analysis is well suited to analyzing communication-related characteristics and social messaging. The PIRANHA team worked towards a graph-centric analysis throughout the program with Neo4j, a high-performance graph database, at the heart of the running system. Using old and synthetic datasets, the team experimented with multiple types of graph analysis: Community/Clique Detection, Nexus Points, and Node2vec. These had varying levels of success at exposing phishing campaign communication patterns, however, there was not enough data at the program-level to fully utilize a graph analysis to combine weak signals (such as the stylometry and outside vetting signals discussed previously) for a decision on friend or foe.

Instead of unifying weak signals into a friend or foe decision using a graph-based classifier, the delivered solution uses a manually tuned Rete rule system. The rule system utilizes a tree-based tracing approach to reach final decisions at the leaves. Asynchronous sources provide new information that can lead to changes in decisions. Specifically, the external vetting and website vetting approaches discussed above can take some time to return with information (as we rate limited our outgoing requests). As new information becomes available, the path through the rule system is updated accordingly.

3.6 Linguistic Signals

3.6.1 Intent Detection

While understanding the intent of an email communication is typically insufficient as a signal for friend/foe decisions in isolation, in the context of active attack detection, intent can serve multiple purposes. First, the intent can be used to anchor an automatic response to the email (in ASED: chatbots) towards information that simultaneously is responsive to the sender's request and elicits relevant content from the sender. For example, an urgent request for funds might make more appropriate a question about bank accounts than an email from an online recruiter. Second, the intent can be used as weak signal that triggers alternative paths in an overall automatic vetting system. Perhaps, for example using the information that sender is introducing themselves as indication that additional vetting of the specific sender should be performed. Third, the intent can be used as information that is provided to the recipient to remind them, e.g., that even though a message displays urgency, they should remain cautious about sharing certain information.

In the first phase of the program, driven by advances in low shot modeling we developed a low-shot approach to detecting email intents. The approach used relied on an underlying transform model (specifically Robustly Optimized Bidirectional Encoder Representations from Transformers (BERT) Pretraining Approach [RoBERTA]) (-large (Liu et al, 2019)) as a mechanism for encoding sentences within a model. We used our own intuitions to manually craft a set of initial positive and negative examples for each intent of interest. Typically, we created a balanced set of 50-100 intents of both positive and negative examples. Crafting the positive examples manually was designed to harness developer intuition for intents that might be relatively rare in available data. We note that a positive example of intent-X can serve as a negative example of intent-Y. Following established practice for using RoBERTa for sentence classification, we encoded each sentence using RoBERTa pretrained weights and fine-tuned the classifier using the positive and negative examples. This results in an initial baseline model. We

hypothesized that our initial model would have insufficient negative examples thus will be prone to over predict a tag. Inspired by active learning, we applied the model to new data (e.g., sentences extracted from emails, samples that we have created) and manually reviewed its positive classification. We augmented the models training with the manually corrected output. This process is intended to reinforce the model's ability to correctly identify some instances of the intent and simultaneously providing hard negative examples that will improve the model's precision. A limitation of the approach is that we are unlikely to stretch the bounds of true positives, thus the model will be subject to data biases in the formulation of the initial positive examples. For example, if sentences with urgent intent always include a few tokens (*e.g., ASAP, urgent, now*) then the model is likely to overfit to these particular vocabulary items. In our own development, we tried to provide broad coverage.

3.6.2 Verifiable Fact Extraction

As described above, the affiliation of an email sender can serve as a key for external vetting. Organizational affiliation has been a long-standing relation target for information extraction and natural language processing. The standard community datasets have focused on extraction in the context of narrative text in which the author describes the affiliation of a third party (e.g., *ACME's CEO Jane Smith).* The language of email is quite different, involving a combination of (a) information in the signature block; (b) explicit introductions by the sender (e.g., *My name is Jane Smith. I'm contacting you from ACME*); (c) implicit relations that link a first- or second-person pronoun to the sender (e.g., *Our work at ACME…*). Here, we assume that the signature block requires independent parsing (either via special purpose regular expressions or by available email processing packages) and explore the explicit and implicit sender-centric relations. Contemporaneous work in information extraction has demonstrated that NLI can be effective for relation extraction tasks. In an NLI context a passage (e.g., "The man, who was walking down the street, purchased a coffee") is combined with a hypothesis that is a templated statement of the relation(s) in question (e.g., "*The man possesses coffee*" (for an ownership relation); "*The man is located in coffee*" (for a located in relation). A general-purpose model labels the passage + hypothesis as *entails, contradicts, other.* An *entailment* label is an assertion by the model that the relation is true based on the passage.

We find the NLI framework appealing because in prior work it has been demonstrated as effective in zero-shot settings (i.e., without task specific training) and because it limits the number of independent components in the natural language processing pipeline. We are able to apply named entity recognition to identify organizations and locations without requiring coreference to cluster mentions of "I," "My," etc. to the sender.

Applying the NLI framework in our context, we are interested in four classes of relations: (1) an employer/organizational affiliate of the sender; (2) the location of the sender, (3) the phone number of the sender, and the (4) the income of the sender. For each of these we define a template (e.g., for employment/affiliation: *The email sender sent this message - <MESSAGE> </s></s> The sender works for <ORG>X).* We apply off-the-shelf named recognition models to identify potential named entities, phone numbers, and incomes in the message. We explored a combination UIUC's OneIE (Lin et al, 2020), HuggingFace's BERT-large named entity recognition (Hugging Face, n.d.[1]) and spaCy's part-of-speech tagging (spaCy, n.d.). We then applied Hugging Face's NLI model (Hugging Face, n.d.[2]) to determine entailment relations.

3.6.3 Data for Is Friend or Foe and Linguistic Features

An ongoing challenge throughout the program was realistic data for sophisticated email attacks. As we describe in Section 4.1, ISI's capabilities were deployed on the program hosted testbed where external groups provided a mix of friend and foe traffic. In general, it was challenging to have access to "realistic" friend traffic for a combination of privacy and experiment setup reasons. For most evaluations, the friend traffic in the testbed was either infrequent or limited to mass-mailing style communication. On the foe-side, in many cases the attacks were difficult to separate from generic cold call emails using language alone.

To address these limitations while still exploring the potential for language-level features to serve as signals of an attack, we explored the creation of a benchmark email corpus that includes manual annotated diverse features that are highlighted as potential signals for phishing attacks. We focused on a mix of signals that can be used for vetting (e.g., the email sender's employer) and signals that users are trained to recognize as signals of phishing. Our research goal here was (1) to determine if these signals could be reliably annotated (in which case, standard supervised machine learning techniques could be used for detection); (2) to establish a new benchmark for testing the detection of such features.

To establish a set of labels we reviewed user facing documentation of anti-phishing guidance (e.g., guides released by the Federal Trade Commission [FTC], n.d.), Google (Google, n.d.), the University of Southern California [USC] Information Technology Services, n.d.). Common themes of these anti-phishing trainings include *asking users to detect urgency, requests to perform some unusual action, requests to share some sensitive information.* Other tell-tale signs of phishing are emails containing links that appear similar to those of reputable sources, but take the user to some obscure Web site, and emails sent from an email address that has a seemingly obscure domain name, similar to a reputable domain name (e.g., g00gle.com instead of google.com). Published user-facing material often provides high-level examples and leaves interpretation (and generalization) to the user. In defining our annotation task, in some cases we attempt to annotate the high-level signal directly-- for example, annotating urgency and requests for passwords directly. For other contexts, we attempted to decompose the signal into pieces that would be simpler to label and eventually detect. For example, one training warns about *"A sender email address that does not match who the email claims to be from"*. To support this class of detection, we seek to identify affiliation claims by the email sender.

Furthermore, we incorporated the annotation of verifiable information to support extended vetting as illustrated in the experiments described above. Verification can include identifying mismatches in:

- The identity established in the text of an email and identifying features in the email metadata. For example, *does the name of the organization match the domain of the actual email address?*
- The identity established in the text of an email and publicly available information about the sender. For example, *does the sender with the given name work in the organization that they claim in the signature?*

To support an annotation workflow, in addition to identifying a set of labels, we identified the span of text that would be annotated for a positive example for each label. Our annotation procedure allowed for four classes annotation spans:

- Annotation of a sentence (e.g., sentence X is *urgent)*
- Annotation of words within a sentence (e.g., *words M-N indicate the employer of a sender*)
- Annotation of the message as a whole
- Annotation an email signature and its parts

Decisions about what span to annotate were tied to our initial insights about the ability of annotators to agree and the potential use of the annotation. For example, *urgency* is a signal, but the word that indicates urgency is unlikely to aid the defense. Thus, annotating urgency at the sentence level seemed an acceptable short-cut over the annotation of *urgent* words. For vetting purposes, the specific email address of the sender is important, thus extracting email from the signature is a distinct annotation task.

Given the challenges of finding real data, we focused on existing sources including the Linguistic Data Consortium's release of Enron Email, other public datasets, and email from the testbed.

3.7 Testbed Management

The program directed performers to work with Kubernetes with Kafka topics for input and output data streams. This meant messages from various channels the program worked over would be parsed, put into a program specific format, and added to an input Kafka topic. From there, each technology could consume the information and act on messages. To match this process, the ISI team produced Friend or Foe results for TA2 teams to consume via a Kafka topic which TA2 teams could consume and produce message response to.

The ISI team worked to build several frameworks to enable technology to be deployed on the testbed, in a local environment and third party environments, while keeping the core of the system unchanged. This simplified the work and testing required to test locally, deploy to the testbed and work with interested third parties.

3.7.1 ISI Local Development and Deployment Setup

For local testing, the ISI team used Minkube to mimic the Kubernetes testbed environment and a custom local networking set up to standup and run a Kafka set up which could take input from files (e.g., emails saved as files) and present these on a Kafka topic the same way input would be received on the program testbed.

This setup enabled quick testing for developers and standardization on a shared suite of input tests before any contributor committed code to the production system. With multiple developers across a wide range of technologies, this testing and standardization was critical for robust code.

3.7.3 Demo and Third Party Deployment Setup

To enable easy demos and to integrate with third parties, ISI created a micro platform which could extract individual signal detections from the testbed system code base and deploy these in

a sandboxed environment via docker-compose. This enabled different hookups for input (e.g., enterprise email systems) and output (e.g., a simple webmail UX for demo purposes).

Early in the program, ISI reached out to multiple companies working in email phishing protection. This included a startup company called Sublime Security. Sublime's model for email protection matched ISI's multiple-signal approach. Specifically, Sublime's goal is to produce a customizable rule set which can express triggers for phishing detection over any aspect of the email including headers, body, and attachments.

ISI provided code to run various signals and work as a "side car" system to Sublime's system. This allowed for asynchronous input and output from the ISI micro platform and enabled Sublime to create rules based on signals from ISI's micro platform output, without holding up Sublime's email processing.

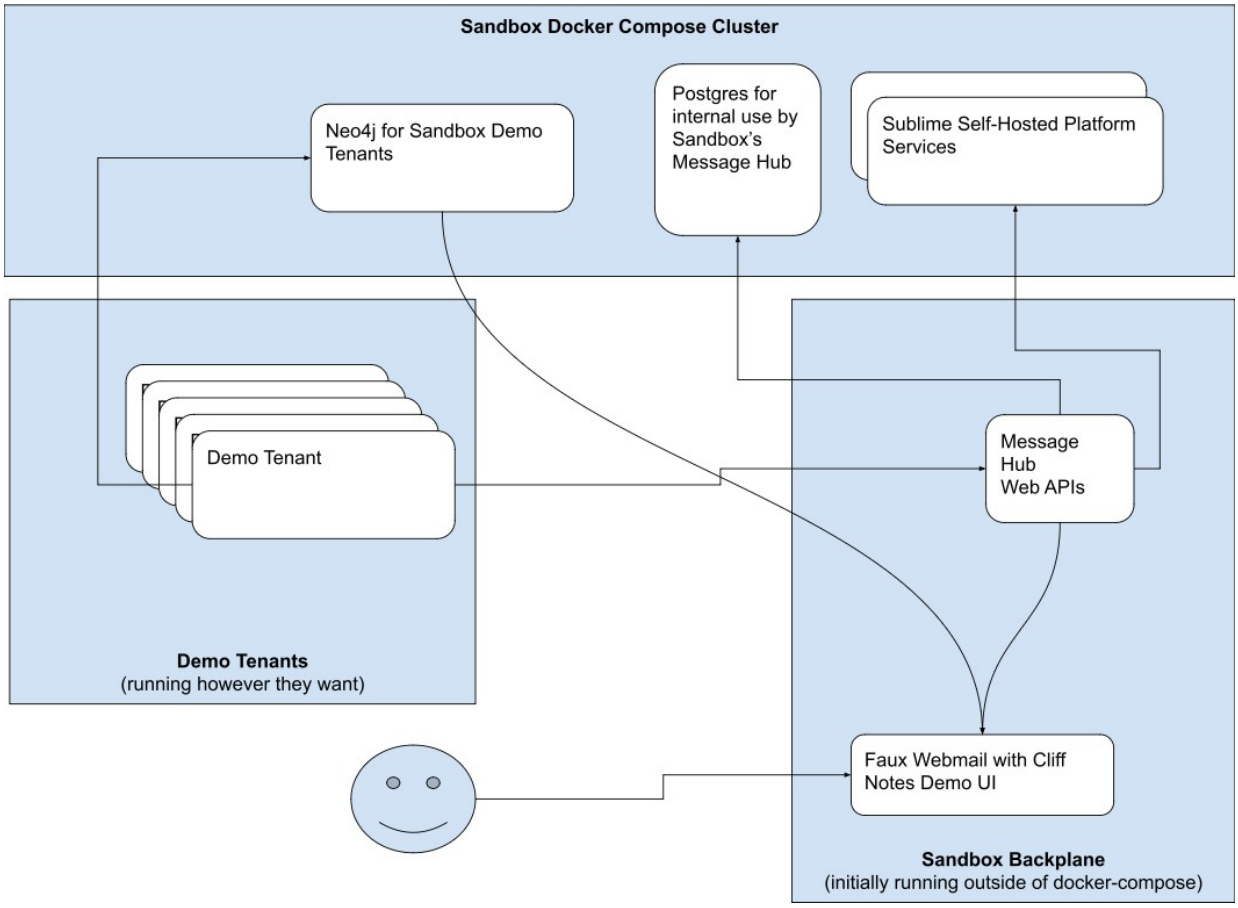

Figure 5: ISI Demonstration and Third Party (Sublime Security) Deployment.

Figure 5 depicts how the micro platform sandbox worked to deploy ISI technologies in a new way and interface with a demo user interface (UI) and third party services such as Sublime Security.

3.7.4 Program Continuous Integration and Deployment Set Up

As part of our team's effort on ASED, we provided CI/CD automation. This effort was headed by our subcontractor at NGC.

CI/CD are force multipliers for a program's execution rhythm. As the saying goes, "*automate all the things!"* The less time your team spends on the mundane and repetitive tasks, the more time they can spend on developing value for the program. A well-orchestrated and streamlined CI/CD process reduces the time-to-production for new features or fixes. Coupled with automated test coverage, CI/CD pipelines enable developers to contribute changes with a greater degree of confidence against adversely affecting the production system.

In Phase 2, the ISI/NGC team developed the following techniques to harmonize the ASED deployment into a single cohesive and well-integrated unit:

- Standardized the entire deployment arrangement on Kubernetes (k8s) by developing and maintaining the canonical k8s deployment descriptors in a repository named "shoal."
- The shoal arrangement could be vertically sliced into TA1 and TA2 minified deployment overlays, enabling each team to work in isolation on their components without needing the entire system.
- Shoal could also be tailored for different production deployment targets and was used towards the end of Phase 2 to support two parallel evaluation testbeds.
- In addition, the team developed the CI/CD pipelines to:
    - Smoke test all merge request changes to shoal by deploying the changes to an ephemeral k8s namespace, and then deploying a suite of integration tests into that same namespace to verify core functionality still worked. Success/failure of the test is fed back into the merge request as a pass/fail flag for review.
    - Automatically deploy all accepted shoal changes (via merge request only) through a "staging" and "prod" promotion pipeline.
    - Automate other axillary operations, such as:
        - Wipe out existing deployments and re-deploy "from clean state".
        - Manage introspection and monitoring of evaluation environment with Prometheus/Grafana by deploying Prometheus operators to automatically discover running services and scrape them for metrics.
        - Keep the documentation site (docs.ased.io) up to date with any/all changes to the various documentation content source repositories.

This effort enabled all teams to seamlessly integrate, test and deploy their solutions on the program testbed.

3.8 Integration and Abstraction

The PIRANHA team developed an API gateway to enable easier transition. All outbound service calls originating from ASED program technologies implemented a basic ASED API, with options to build out additional calls. The PIRANHA team built an API-gateway, which routed all requests in and out of every ASED technology to the appropriate endpoint adapter. The endpoint adapter(s) behind the gateway were responsible for implementing the service contracts. The PIRANHA team worked with program requirements and each team's requirements to define and Enterprise API contract. This API defined all the available service endpoints along with payload schemas for each incoming and outgoing message. This gave the ASED subsystems an internally stable API to target, independent of the deployment environment.

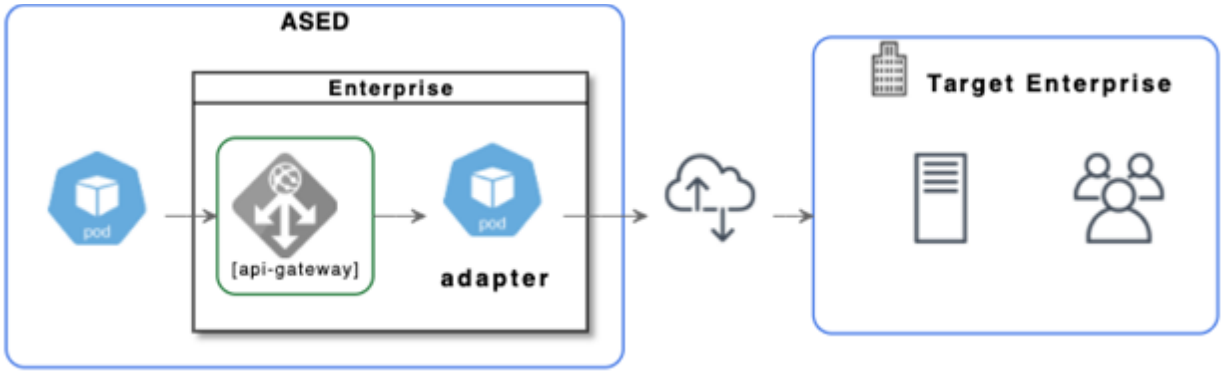

Figure 6: Abstraction Enabling ASED Subsystems to be Ported to Different Target Enterprise Environments without Needing to Modify Internal Communication.

4.0 RESULTS AND DISCUSSION

Our system performed well throughout evaluation, achieving 100% recall and 99.9% precision for friend or foe in all three campaigns in the final program evaluation. There was a single non-Test & Evaluation (T&E) email which was marked as foe. This email, under manual inspection, appeared to be spam or, at the very least, was a "cold call" style email where a stranger reached out to someone to offer an opportunity. Differentiating between spam, scams, credential phishing and more targeted phishing in communication will remain a challenge as even experts do not fully agree on what distinguishes these categories from each other. In the following sections, we first describe in more detail performance in the official program testbed and then describe internal results for several components that we developed in our exploration of phishing detection.

4.1 Program Evaluations

Program evaluations were run using a program-wide testbed at several points during the program. For the evaluation, systems needed to be deployed and active on the program's Kubernetes infrastructure. During program evaluations incoming email was fed to the deployed system via a Kafka and system decisions of friend or foe were fed to a central database for scoring and tracking. Incoming traffic during these evaluations was a mix of emails to and between consenting participants (in program terms "the cohort") recruited by the test and evaluation team, mailing lists, and traffic from the red team run by the test and evaluation team. For official metrics, red team traffic was treated as foe, all other traffic (including e.g., spam to a mailing list) was treated as friend. The composition of the cohort varied between evaluation periods. In Phase 1, the cohort included Jet Propulsion Laboratory (JPL) employees. In Phase 2, in some evaluations the cohort was composed of Virginia Tech students, in others the cohort was of Data Machine Corp (DMC) employees, and finally in some cases the cohort was empty, and all traffic was mailing list derived.

Using the continuous deployment approach described in Section 3.7.1, PIRANHA's detection system was live with minimal disruption throughout all program evaluations. Table 1 presents results on the core phishing detection task for various evaluation cycles.

Table 1: Primary Detection Results for Program Evaluations.

	Is Friend or Foe (F1)	Foe False Alarms (%)
Phase 1	96.6	2.7
2021-Combined	96.6	1.8
Winter2021-VTNatSec	96.7	0.6
Winter2021-DMC	23.6	8.1
Fall2022-Combined	99.9	0.0

The Winter 2021 cycle with the DMC cohort stands out as having a particularly high false alarm rate and a particularly low F1 for the Is Friend or Foe detection task. Because many features of friend and foe emails overlap, our detection system is designed to incorporate a safelist that relies on message metadata to decrease the likelihood of attacks for trusted senders. The safelist feature includes both organizational information (e.g., *trust senders that share my domain)* and user

specific information. The safelist is also setup such that it can be pre-initialized for an organization and also learn-on fly given observed traffic and user feedback. In the context of the DMC cohort, we found that the learning process had not been initialized which resulted in overly aggressive foe detection. In further analysis, we simulated two types of safelist learning. The first relies on learning to trust specific email senders (e.g., allowing jsmith@isi.edu), the second relies on safelists for domains as a whole (e.g., allowing all of isi.edu). Table 2 presents these results. For this set of experiments, we treated the time-period of 10 – 20 January as the test period. During that ten-day window, there were 763 messages, 24 of which were foe. Given the small number of foe messages, we present raw true positive (TP) and false alarm (FA) results rather than precision and recall. In the particular evaluation, neither strategy yielded a reduction in foe true positives, i.e., neither email address nor domain safelists decreased the effectiveness of blocking foe traffic. With as little as a six-day training and a conservative email-address based safelist strategy, we saw a 35% reduction in FAs. With a more aggressive domain-based strategy, we saw a 63% reduction with the six-day window. A longer training window (still less than a month) further reduces the FA rates. These results point to the importance of active detection whereby the system learns about organization and/or individual it is protecting.

Table 2: Reduction in Foe FAs with Different Training Periods and Different Approaches to Learning a Safelist for the Winter 2021 DMC Dataset.

Domain Adaptation Data			**Adaptation: Learn Email Addresses as Friend**			**Adaptation: Learn Email Domains as Friend**		
Training Period	***# Days Training***	***# Messages During Training***	***Training: # Unique addresses***	***TP: Foe***	***FA: Foe***	***Training: # Unique Domains***	***TP: Foe***	***FA: Foe***
None	0	0	0	19	49	0	19	49
14 Dec. – 20 Dec.	6	530	116	19	32	69	19	18
14 Dec. - 27 Dec.	13	725	139	19	32	81	19	18
14 Dec. -3 Jan.	20	979	158	19	31	96	19	17
14 Dec. – 9 Jan.	26	1647	206	19	11	121	19	2

In addition to producing friend/foe labels, the PIRANHA system contributed to the flags that were assessed as extractable information about attackers. In the evaluation setup, to complement TA2's interactive flag detection through dialogue, PIRANHA focused on the static extraction of first name and last name from email metadata. While portions of email metadata are human readable text, the content is typically formatted in a manner that is ill-suited for standard natural language processing techniques, and more suitable for structured extraction with dictionaries. Here we focused on the first name and surname often associated with the sender's metadata in a format determined by their mail client (e.g., *Jo Smith <jsmith@isi.edu>*, or *Smith, Jo <jsmith@isi.edu>*). This capability was deployed in the final evaluation. Unfortunately, in the final evaluation cycle, there were only a small number of distinct foe senders and for two of those senders, each of which sent many messages, the dictionary-based approach yielded false alarms. Figure 7 shows these two errors and one success. At the core of the errors is the potential

polysemy of person names with non-person name words. For example, Virginia is both a common first name and the name of a state. While less common both Clerk and Commission are possible surnames. To improve the precision of metadata-based flag detection would require more complex dictionaries and potentially organization specific stop-lists.

- benefits@thevt.tech: Virginia Commission ✗
  - Reported 175 times
  - Actual name is 'Sora' found after much questioning

- s.suzuki@thevt.tech: Virginia Clerk ✗
  - Reported 41 times
  - Actual name is 'Sora Suzuki' found after questioning

- mmiacuevas@gmail.com: Mia Cuevas ✓
  - Reported 22 times
  - Found from Email Address field

Figure 7: First Name, Last Name Examples from Final Evaluation.

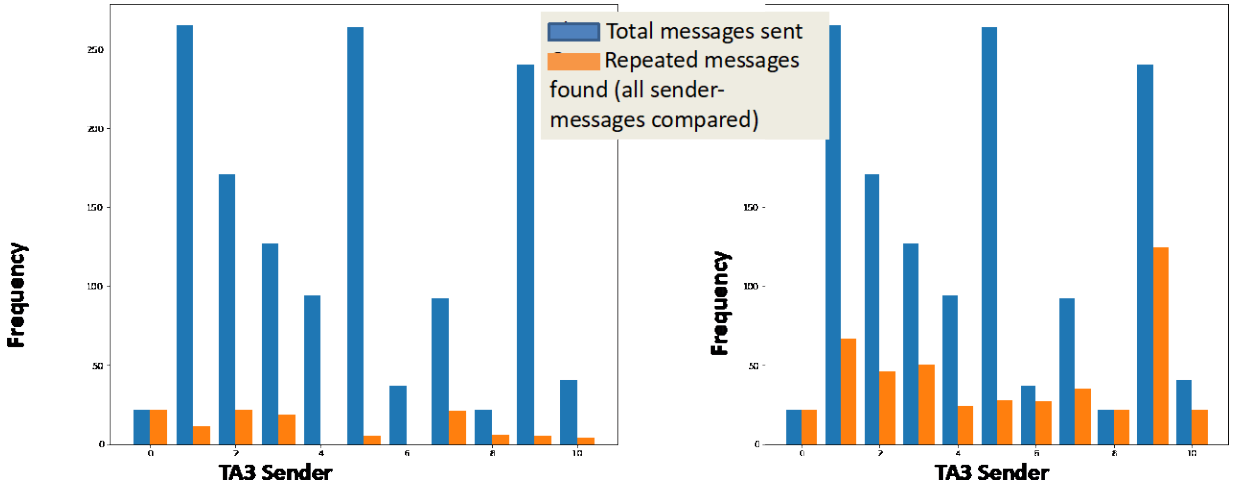

Figure 8: Frequency of Identical(left) and/or Similar(right) Foe Messages.

The degree to which we were able to use the testbed data for more in-depth analysis depended on the scope and data realism of the testbed. To understand this, for the final evaluation we looked at scope and variation of the foe messages. The fact that a handful of senders yielded many messages became apparent in the flag analysis where 175 of the foe messages were from one sender. Furthermore, that messages sent by a sender were often identical or very similar. While there were 1,375 red team messages sent during the evaluation, 115 of the messages (8%) were repeated in identical form. When we applied a slightly more lenient fuzzy match to the messages, we found that 434 (34%) were repeated. Figure 8 shows per-sender total messages (blue bar) and frequency of messages (orange bar) using both exact match (left) and fuzzy match(right). While this approach of repeated messages is common in large-scale phishing campaigns and the repeated messages test system scalability, repeats of the same message will be processed by the

phishing detection in the same way and thus do not improve the breadth of phenomena measured.

4.2 URL and Website Vetting

To test vetting linked websites, we built a collection system to download phishing sites just after they were discovered—and reported to OpenPhish (OpenPhish, n.d.). We also collected 1,000 "good" websites ranging from login pages to randomly selected pages from the top 1,000 most popular sites to sites randomly selected. We used this dataset for training and testing our approach.

We demonstrate that the classifier is able to correctly differentiate phish vs friendly(ham) pages in a realistic mix of pages with 99.3% accuracy, with a very low false positive rate of 0.1%, making it on par with other automated phish-detecting systems. When vetting sites, both inter-domain and intra-domain features were the dominating factor in importance. The reduction of false positives is primarily achieved using inter-domain and intra-domain features.

A key part of our approach is in drawing features from website structure that would be costly for an attacker to change when creating a spoof site (as discussed in the previous section). We demonstrate the ubiquitous-ness of these hard-to-change features by examining the dominating features in our detection approach. Figure 9 shows the results of detection when each feature is removed. Features which are dominant in detection (such as external and internal linking strategies between sites), reduce detection accuracy when removed. This demonstrates that intra- and inter-linking between spoof websites and legitimate websites (*_d and *_f) are common and dominating features in detecting phishing sites.

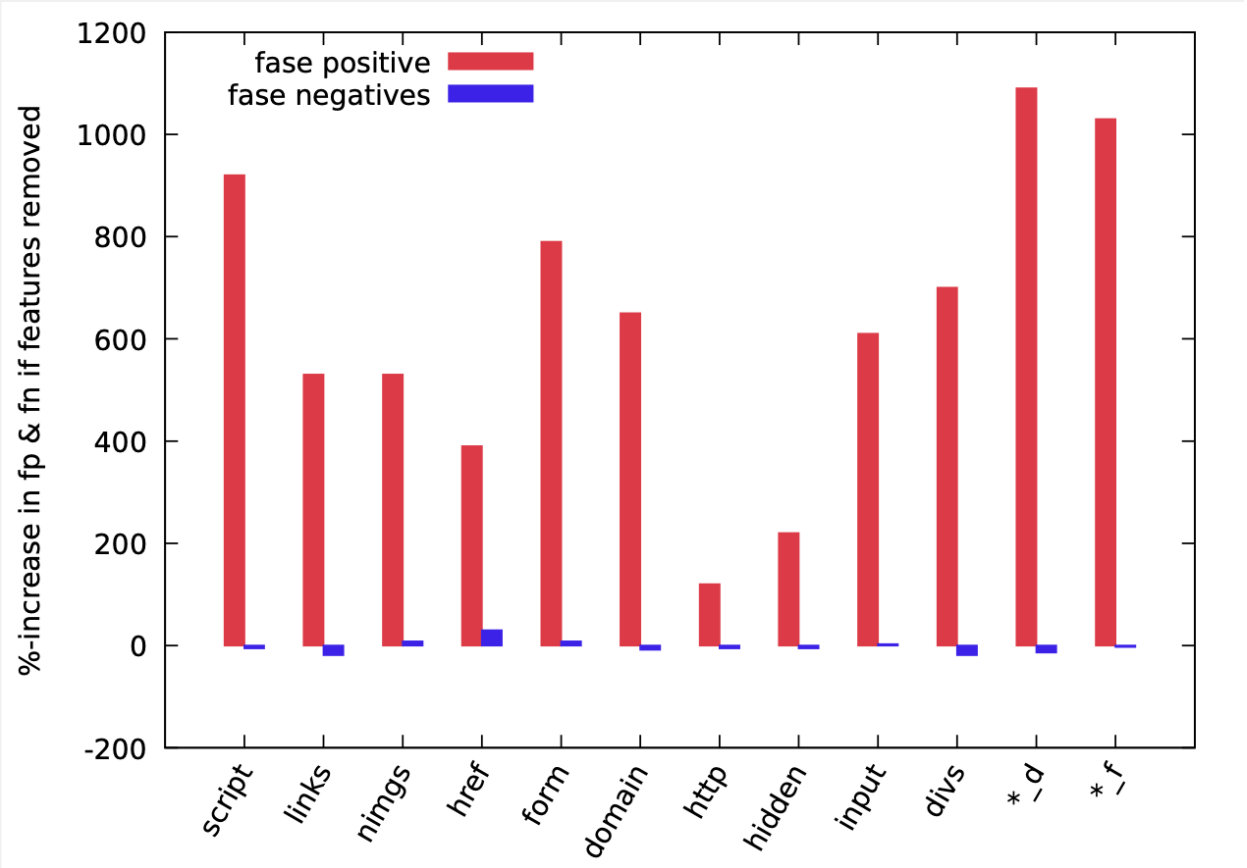

Figure 9: Percent Increase in False Positives and False Negatives in Website Classifier Testing when Features are Removed.

4.3 Stylometry in Phishing

To test stylometry on data from the T&E team on the program, we selected samples from ten email addresses associated with the T&E team and five email addresses not associated with the T&E team. The addresses were selected based on the number of emails they sent. By utilizing indicative stylometry, it was possible to group authors and distinguish the "style" of T&E emails from background or "good" messages. For a more realistic testing scenario, data from the SRI Pilot Study (SRI was a separate performer on the program who used the phishing emails SRI received as a way to test the program's dialogue engines) was used to examine device fingerprints and meta-data as a basis for ground truth. Unfortunately, the fingerprints from live-phishing emails appeared to come from containerized environments, and as such, these fingerprints rotated as the phisher engaged with the program's automated dialogue systems. Without ground truth on which phishers were consistently the same author (within an email thread or across threads), testing our stylometry approach to cluster authors or phishing campaigns was not possible.

4.4 Linguistic Analysis

4.4.1 Intent Classification

As described in Section 3.6.1, we explored a fine-tuned transformer-based approach to building intent classifiers. While the intents of interest have the potential to provide valuable signals in active detection, they are often rare in email data. Thus, while the approach was integrated into the overall deployed system, we also performed internal measurement of the classifiers by developing a small per-intent test set of positive and negative examples. These results were intended as a means of testing the general approach on intents with different characteristics. Specifically, we developed training and evaluated for: *requests to follow a link, requests for fundraising, introductions of the sender to the recipient (e.g., Hi, I'm.....), requests for phone calls, specific recruiter traffic, and requests to schedule meetings*. The intents were selected to cover cases that were discussed as high value in program meetings.

Table 3 below presents F1 using four different models as the base of the fine-tuning process.

Table 4 presents precision and recall for the same set of intents and models. Recruiting appeared to be particularly challenging to identify, suggesting perhaps that recruiting is confused with other classes. In general, RoBERTa large outperforms the BERT base models which aligns with experiments in other domains. Here in particular, the inclusion of both web-liked data may yield a better pre-trained feature space. There does not seem to be a strong trend with respect to whether this approach yielded high precision or high recall models. We suspect that in real applications, it would be useful to apply the data-driven approach to finding new positive and negative emails to the specific traffic of interest. Unfortunately, traffic in the program testbed was insufficient for running such experiments.

Table 3: Performance as F1 on Small Development Set for Intents of Interest.

	bert-base-uncased	bert-base-cased	roberta-base	roberta-large
	F1	F1	F1	F1
follow_link	0.81	0.7	0.81	1
fundraising	1	1	1	0.93
introduction	0.4	0.71	0	0.79
phone_call	0.78	0.69	0.81	1
recruiter	0	0.57	0.6	0.5
scheduling	0.93	0.95	0.9	0.88

Table 4: Precision (Prec) and Recall of Intents of Interest on Small Internal Development Set.

	bert-base-uncased		bert-base-cased		roberta-base		roberta-large	
	Prec	Recall	Prec	Recall	Prec	Recall	Prec	Recall
follow_link	0.72	0.92	0.54	1	0.84	0.78	1	1
fundraising	1	1	1	1	1	1	1	0.88
introduction	1	0.25	0.83	0.62	0	0	0.85	0.75
phone_call	0.6	1	0.9	0.56	0.71	0.93	1	1
recruiter	0	0	0.5	0.66	0.42	1	0.4	0.66
scheduling	0.86	1	0.9	1	0.86	0.95	0.8	1

4.4.2 Verifiable Fact Extraction

We integrated the verifiable fact extraction models into the integrated, testbed phishing detection engine. However, we found the contexts of interest to be sufficiently rare in the testbed that we were unable to calculate test-bed specific measures. Before integration, we performed an internal analysis of the accuracy of the approach. We the extraction approach over a small internal development set of email messages. We judged the accuracy of the entity extraction (i.e., what percentage of the identified entity had the correct type), and then assessed, given those extractions, the precision and recall of the relation extraction component. This approach to measurement, over-estimates recall because it does not account for extents where the entity extraction fails to identify a context. Results of this assessment appear in Table 5. Overall, the combined entity extraction method achieved high accuracy for all entity types. In our data set, perhaps not surprisingly, there were no positive examples of income. Recall for the natural language inference-based approach was reasonably high for the three types of relations in the dataset. However, its precision was low for organization/affiliation.

Table 5: Assessed Accuracy of Entity Extraction and Assessed Recall, Precision, and F1 of Sender-Entity Relations.

Type/Relation	Entity Accuracy (%)	Precision	Recall	F1
Organization/Affiliation	96	0.4	0.9	0.6
Location/Located In	84	0.8	0.8	0.8
Phone Numbers/--	99	1	1	1
Incomes/--	94	0	-	-

In addition to measuring performance of this approach, we prepared a standalone demonstration that highlighted both extracted entities and entity-sender relations in an email. Figure 10 shows this demonstration capability on a sample email.

Figure 10: Demonstration of Relation Annotated Email.

4.4.3 Data for Is Friend or Foe and Linguistic Features

We performed dual annotation on 927 emails from a mix of sources: 323 emails from the Enron email corpus; 174 from Kaggle's fraudulent email corpus (https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus), and 430 emails from the program testbed. Table 6 presents the number of positive instances of each label class in the annotated data set organized by annotation scope (as described previously). In many cases, despite the large corpus, the labels were infrequent. For example, messages that are sent on behalf of an organization represent less than 20% of the messages annotated. The sparsity of positive labels is even more pronounced for sentence level scopes. The corpus has over 10,000 sentences, many labels appear less than 100 times (i.e., less than 1% of the data).

Table 6: Counts of Positive Instances of Annotated Labels, with Cohen's Kappa as a Measure of Agreement for the More Common Labels.

Scope	Description	Cohen's Kappa	Count
Message	Message from a person asking for something	1	594
	Message in which a person explicitly represents an organization as whole	0.98	195
	Message that is sent by an organization (not an individual)	0.92	184
Sentence	Mention of attachment		36
	Request to click on a link	0.87	144
	Introduction by sender		97
	Offering money	0.96	299
	Request for phone call	0.96	108
	Offer of a product	0.98	111
	Recruitment (typically for employment)	0.98	33
	Request for a meeting/scheduling information	0.97	151

	Offer of service		69
	Unsubscribe link		20
	Request to use a specific external product officially used by the sender's organization (e.g. zoom)		36
	Request for password		26
	Polite tone	0.53	352
	Urgent tone	0.53	199
	Uncommon URLs (as determined by an external list)	0.98	138
	URL that is unrelated to the sender's organization		59
Signature	Full signature block	1	173
	Physical address		22
	Email address		28
	Full name	1	133
	Job Title		32
	Represented organization	1	89
	Phone number	1	52
	Signoff	1	89
	URL		9
	Social medial handle		0
Word/ Phrase	Mention of recipient's affiliated organization		0
	Mention of sender's geographical location	0.97	128
	Mention sender's affiliated organization		44

The creation of this dataset involved iterative rounds of revising the guidelines to try to develop a common sense of this labels. Our annotation context was atypical— annotators worked closely together in early iterations of annotation. This may have led to us achieving higher levels of agreement than we would see if the task were assigned to new annotators. Despite this, we include agreement (as Cohen's Kappa) for the more common classes in Table 6.

Agreement for several of these classes is high, the two major exceptions being tone-related content (politeness, urgency). Notably, some of this content (e.g., *signatures*) takes a stereotyped form that we would expect people to easily identify. Fine-grained labels (e.g., *a request for a phone call, an offer of a product*) seem to yield higher agreement. Intuitively, this aligns with our prior experience in more classic information extraction tasks. Agreement has typically been higher for clearly definable entities (e.g., *person names*) than for more abstract constructs (e.g., *the linguistic indicators for a transportation event*).

While we were unable to develop and test models against this annotated corpus, we believe the annotation and summary of the annotation can serve as useful springboard in the future. For some labels (e.g., *phone numbers, URLs),* standard approaches are already readily available and even commercially deployed. In these cases, the concrete annotation illustrates the challenge of data sparsity when producing a sufficiently large dataset for benchmarking. For other labels, there is overlap with existing natural language processing tasks, but its application to an email

benchmark is new. Here again, the annotation illustrates the challenge of sparsity in email as well providing examples in how these labels manifest themselves informal text. In certain cases, for example the detection of an employment relation in the context of relating to the sender of the message (rather than all employment relations), the most useful definition for email foe identification is different than that of the standard natural language processing task. This initial dataset provides a starting place for identifying these differences.

4.5 Testbed Management

Harmonizing the various components and teams into a canonical deployment with shoal proved to be helpful in standardizing the deployment currency each team brought to the table. We no longer had to deal with all the unique scripts and deployment packages.

Concourse (concourse-ci.org) was introduced and utilized as the CI/CD engine that ran continuously throughout the entirety of Phase 2. It became the centerpiece for automating all the interactions across the engineering landscape (Figure 11).

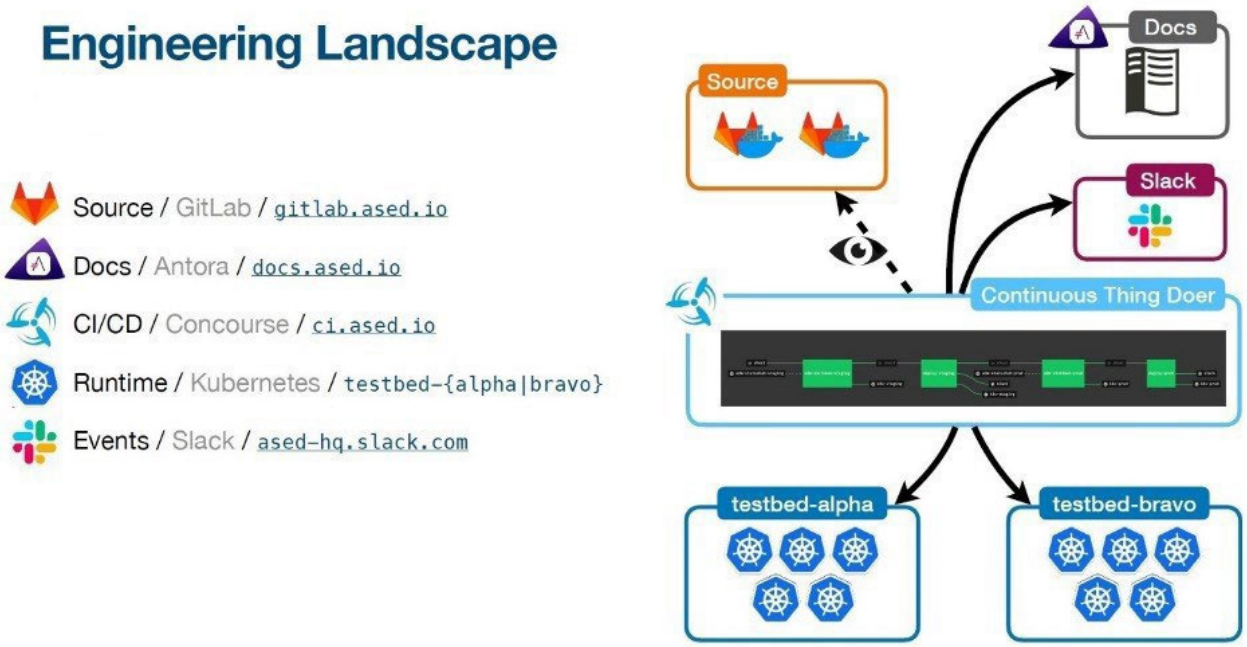

Figure 11: Managing CI/CD Pipelines for System Deployment on the Program Testbeds.

Whenever a change in shoal was observed, a Concourse pipeline would initiate a deployment of those changes to our staging namespace. If all went well in staging, those same changes would then be promoted to our production namespace which was the deployment under evaluation continuously. Figure 12 illustrates this across two different yet concurrent deployment evaluation environments.

Lastly, one of the challenges with multiple concurrent deployments is keeping an eye on everything. ASED utilized Prometheus to scrape for metrics, and Grafana for displaying those metrics with savvy dashboards. However, once we started operating more than one observable

deployment, we also incurred multiple Prometheus/Grafana instances to monitor them. This led to needing to create and manage dashboards across multiple distinct instances.

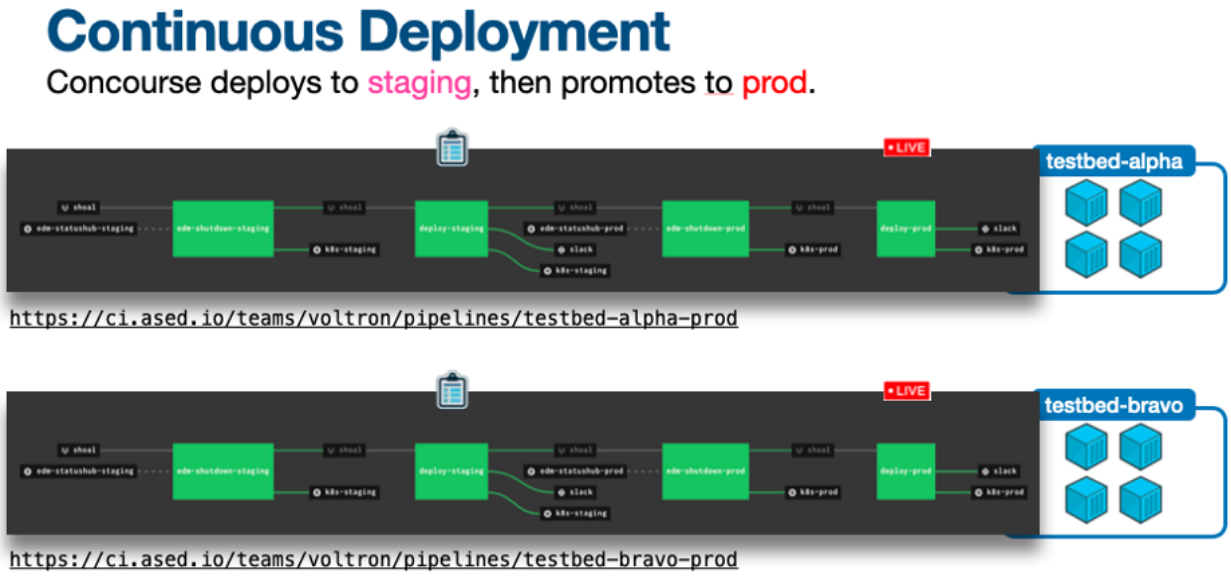

Figure 12: Monitoring Continuous Deployment Pipelines.

A Concourse pipeline was written to automate the synchronization of the dashboard content. ASED team members could make a change to their dashboard layouts and widgets in one Grafana instance, and Concourse would automatically sync those changes to the other Grafana instance(s). Figure 13 illustrates these dashboards. This ensured our monitoring dashboards rendered the same type of information from all currently deployment instance(s) of ASED, even in the face of user changes to the dashboard widgets and layout.

Figure 13: Synced Grafana Dashboards.

4.6 Integration and Abstraction

In an endeavor to obtain more data, the PIRANHA team partnered with a startup company (Sublime Security), who was producing a rule-based system to enable a highly customizable filtering approach to scams and phishing for enterprise email. The PIRANHA team was able to modify the API gateway designed for the program testbed and use this structure to integrate with the start up's rule-based approach.

5.0 CONCLUSIONS

The PIRANHA team identified several features which provide a weak signal indicator that a message is phishing. These signals, on their own, aren't necessarily enough to make an automatic decision, but combined, these can indicate phishing with some certainty. The PIRANHA team also identified methods to utilize neural dialogue in a more directed fashion, which is appropriate for situations where an attacker is unwilling to interact with bots or interested in acting maliciously with automated dialogue.

Phishing, and social engineering attacks in general, will continue to be a problem moving forward. Black-box corporate solutions and user training are the gold standard for defense currently. We anticipate that some solutions will need to be deployed at the end-user level to fully combat the problem as many indicators of stealthy phishing attacks are not typically enough context to block a message. Calling out these indicators to an end-user can augment user anti-phishing training.

The weak signals identified and developed by the PIRANHA team can also be useful at the organizational level to make friend or foe decisions but will require further work with real-world traffic to perfect how these signals are incorporated to make automated decisions.

Our CI/CD pipelines brought consistency and stability to Phase 2, while reducing the cognitive burden on each team to maintain a persistent deployment presence. Without this effort in Phase 2, the disjointed chaos from phase 1 would have hampered the efficacy of Phase 2.

6.0 REFERENCES

FTC (n.d.). Phishing. Retrieved April 1, 2023. https://www.ftc.gov/business-guidance/small-businesses/cybersecurity/phishing.

Google (n.d.). Avoid and report phishing emails. Retrieved April 1, 20203. https://support.google.com/mail/answer/8253?hl=en

Hugging Face. (n.d.[1]). BERT Large NER. Retrieved April 1, 2023. https://huggingface.co/dslim/bert-large-NER

Hugging Face. (n.d.[2]). RoBERTa Large MNLI. Retrieved April 1, 2023. https://huggingface.co/roberta-large-mnli

Hyundong Cho, Genevieve Bartlett, and Marjorie Freedman. (2021). Agenda Pushing in Email to Thwart Phishing. In *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*, pages 113–118, Online. Association for Computational Linguistics.

Li, Y., Qian, K., Shi, W., & Yu, Z. (2020). End-to-End Trainable Non-Collaborative Dialog System. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(05), 8293-8302. https://doi.org/10.1609/aaai.v34i05.6345

Lin, Y., Ji, H., Huang, F. and Wu, L., 2020, July. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7999-8009).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint* arXiv:1907.11692.

spaCy. (n.d.). Entity Recognizer. Retrieved April 1, 2023. https://spacy.io/api/entityrecognizer

OpenPhish, (n.d.). Phishing Intelligence. Retrieved April 1, 2023. https://openphish.com/

USC Information Technology Services. (n.d.) Phishing: Don't Take the Bait! Retrieved April 1, 2023. https://itservices.usc.edu/security/phishing/

Ward, M., & Subramnian, S. (2022). Cybersecurity Study. 2022 Deloitte–NASCIO Cybersecurity Study. https://www.deloitte.com/an/en/our-thinking/insights/industry/government-public-services/2022-deloitte-nascio-study-cybersecurity-post-pandemic.html

Yahoo! Japan. (2020). Neighborhood Graph and Tree for Indexing High-dimensional Data. Github: Nearest Neighbor Search with Neighborhood Graph and Tree for High-dimensional Data. https://github.com/yahoojapan/NGT

7.0 SYMBOLS, ABBREVIATIONS, AND ACRONYMS.

ACE	Automated Commercial Environment
API	Application Programming Interface
ASED	Active Social Engineering Defense
BERT	Bidirectional Encoder Representations from Transformers
CD	Continuous Deployment
CI	Continuously Integrated
DARPA	Defense Advanced Research Projects Agency
DMC	Data Machines Corp
DNS	Domain Name System
FA	False Alarm
FST	Finite state transducers
FTC	Federal Trade Commission
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ISI	Information Sciences
JPL	Jet Propulsion Laboratory
k8s	Kubernetes
NGC	Northrup Grumman Corporation
NLI	Natural Language Inference
PIRANHA	Preventing Information Removal and Nabbing Harmful Actors
RoBERTA	Robustly Optimized BERT Pretraining Approach
SE	Social Engineering
SM	Social Media
SMS	Short Message Service
SSL Certs	Secure Sockets Layer certificates
T&E	Test & Evaluation
TA	Technical Area

TP	True Positive
UI	User Interface
URL	Uniform Resource Locator
USC	University of Southern California