Proceedings of the Workshop on Social Threats in Online Conversations: Understanding and Management (STOC-2020), pages 1–8
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Active Defense Against Social Engineering: The Case for Human Language Technology

Adam Dalton, Ehsan Aghaei, Ehab Al-Shaer, Archna Bhatia, Esteban Castillo, Zhuo Cheng, Sreekar Dhaduvai, Qi Duan, Bryanna Hebenstreit, Md Mazharul Islam, Younes Karimi, Amir Masoumzadeh, Brodie Mather, Sashank Santhanam, Samira Shaikh, Alan Zemel, Tomek Strzalkowski, Bonnie J. Dorr
IHMC, FL {adalton,abhatia,bmather,bdorr}@ihmc.us
SUNY Albany, NY {sdhaduvai,bhebenstreit,amasoumzadeh,ykarimi,azemel}@albany.edu
UNCC, NC {eaghaei,ealshaer,zcheng5,qduan,mislam7,ssantha1,sshaikh2}@uncc.edu
Rensselaer Polytechnic Institute, NY {castie2,tomek}@rpi.edu

Abstract

We describe Panacea, a system that supports natural language processing (NLP) components for active defenses against social engineering attacks. We deploy a pipeline of human language technology, including Ask and Framing Detection, Named Entity Recognition, Dialogue Engineering, and Stylometry. Panacea processes modern message formats through a plug-in architecture to accommodate innovative approaches for message analysis, knowledge representation and dialogue generation. The novelty of the Panacea system is that uses NLP for cyber defense and engages the attacker using bots to elicit evidence to attribute to the attacker and to waste the attacker's time and resources.

1 Introduction

Panacea (Personalized AutoNomous Agents Countering Social Engineering Attacks) actively defends against social engineering (SE) attacks. *Active* defense refers to engaging an adversary during an attack to extract and link attributable information while also wasting their time and resources in addition to preventing the attacker from achieving their goals. This contrasts with *passive* defenses, which decrease likelihood and impact of an attack (Denning, 2014) but do not engage the adversary.

SE attacks are formidable because intelligent adversaries exploit technical vulnerabilities to avoid social defenses, and social vulnerabilities to avoid technical defenses (Hadnagy and Fincher, 2015). A system must be socially aware to find attack patterns and indicators that span the socio-technical space. Panacea approaches this by incorporating the F3EAD (Find, Fix, Finish, Exploit, Analyze, and Disseminate) threat intelligence cycle (Gomez, 2011). The *find* phase identifies threats using language-based and message security approaches.The *fix* phase gathers relevant and necessary information to engage the adversaries and plan the mitigations that will prevent them from accomplishing their malicious goals. The *finish* phase performs a decisive and responsive action in preparation for the *exploit* phase for future attack detection. The *analysis* phase exploits intelligence from conversations with the adversaries and places it in a persistent knowledge base where it can be linked to other objects and studied additional context. The *disseminate* phase makes this intelligence available to all components to improve performance in subsequent attacks.

Panacea's value comes from NLP capabilities for cyber defense coupled with end-to-end plug-ins for ease of running NLP over real-world conversations. Figure 1 illustrates Panacea's active defense in the form of conversational engagement, diverting the attacker while also delivering a link that will enable the attacker's identity to be unveiled.

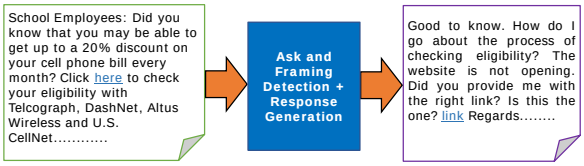

Figure 1: Active Defense against Social Engineering: Attacker's email (left) yields bot's response (right)

1.1 Use Cases

Panacea's primary use cases are: (1) monitoring a user's inbox to detect SE attacks; and (2) engaging the attacker to gain attributable information about their true identity while preventing attacks from succeeding. Active SE defense tightly integrates offensive and defensive capabilities to detect and respond to SE campaigns. Engaging the adversary uniquely enables extraction of indicators required to confidently classify a communication as mali-

cious. Active defenses also carry significant risk because engagement can potentially harm an individual's or organization's reputation. Thus, high confidence classification is vital.

1.1.1 Monitoring and Detection

Panacea includes an initial protection layer based on the analysis of incoming messages. Conceptual users include end users and IT security professionals. Each message is processed and assigned a label of friend, foe, or unknown, taking into account headers and textual information of each message. The data obtained from this analysis is converted into threat intelligence and stored in a knowledge graph for use in subsequent phases, e.g., for meta analysis and message analysis in a broader context within a thread or in similar messages delivered to multiple users.

1.1.2 Engagement and Attribution

Passive defenses are finished once a threat is discovered, defused, and deconstructed; at this point Panacea's active defenses become engaged. Panacea's active defenses respond to the attacker's demands, reducing the risk that the attacker will catch on that they've been fingered. As such, any requests made by Panacea are more likely to be fulfilled by the attacker, bulwarked by hopes of eventual payoff. Such requests are implemented as a collection of flag seeking strategies built on top of a conversational theory of *asks*. Flags are collected using information extraction techniques. Future work includes inferential logic and deception detection to unmask an attacker and separate them from feigned identities used to gain trust.

2 Related Work

Security in online communication is a challenge due to: (1) attacker's speed outpacing that of defenders to maintain indicators (Zhang et al., 2006); (2) phishing site quality high enough that users ignore alerts (Egelman et al., 2008); (3) user training falling short as users forget material and fall prey to previously studied attacks (Caputo et al., 2013); the divergent goals of the attacker and defender (Li et al., 2020); and (4) defensive system maintainers who may ignore account context, motivations, and socio-economic status of the targeted user (Oliveira et al., 2017). Prior studies (Bakhshi et al., 2008; Karakasilitiosis et al., 2006) demonstrate human susceptibility to SE attacks. Moving from bots that detect such attacks to those that produce "natural sounding" responses, i.e., conversational agents that engage the attacker to elicit identifying information, is the next advance in this arena.

Prior work extracts information from email interactions (Dada et al., 2019), applies supervised learning to identify email signatures and forwarded messages (Carvalho and Cohen, 2004), and classifies email content into different structural sections (Lampert et al., 2009). Statistical and rule-based heuristics extract users' names and aliases (Yin et al., 2011) and structured script representations determine whether an email resembles a password reset email typically sent from an organization's IT department (Li and Goldwasser, 2019). Analysis of chatbot responses (Prakhar Gupta and Bigham, 2019) yields human-judgement correlation improvements. Approaches above differ from ours in that they require extensive model training.

Our approach relates to work on conversational agents, e.g., response generation using neural models (Gao et al., 2019; Santhanam and Shaikh, 2019), topic models (Dziri et al., 2018), self-disclosure for targeted responses (Ravichander and Black, 2018), topic models (Bhakta and Harris, 2015), and other NLP analysis (Sawa et al., 2016). All such approaches are limited to a pre-defined set of topics, constrained by the training corpus. Other prior work focuses on persuasion detection/prediction (Hidey and McKeown, 2018) but for judging when a persuasive attempt might be successful, whereas Panacea aims to achieve effective dialogue for countering (rather than adopting) persuasive attempts. Text-based semantic analysis is also used for SE detection (Kim et al., 2018), but not for *engaging* with an attacker. Whereas a bot might be employed to warn a potential victim that an attack is underway, our bots communicate with a social engineer in ways that elicit identifying information.

Panacea's architecture is inspired by state-of-the-art systems in cyber threat intelligence. MISP (Wagner et al., 2016) focuses on information sharing from a community of trusted organizations. MITRE's Collaborative Research Into Threats (CRITs) (Goffin, 2020) platform is, like Panacea, built on top of the Structured Threat Intelligence eXchange (STIX) specification. Panacea differs from these in that it is part of operational active defenses, rather than solely an analytical tool for incident response and threat reporting.

3 System Overview

Panacea's processing workflow is inspired by Stanford's CoreNLP annotator pipeline (Manning et al., 2014a), but with a focus on using NLP to power active defenses against SE. A F3EAD-inspired phased analysis and engagement cycle is employed to conduct active defense operations. The cycle is triggered when a message arrives and is deconstructed into STIX threat intelligence objects. Object instances for the identities of the sender and all recipients are found or created in the knowledge base. Labeled relationships are created between those identity objects and the message itself.

Once a message is ingested, plug-in components process the message in the *find* phase, yielding a response as a JSON object that is used by plug-in components in subsequent phases. Analyses performed in this phase include message part decomposition, named entity recognition, and email header analysis. The *fix* phase uses components dubbed *deciders*, which perform a meta-analysis of the results from the *find* phase to determine if and what type of an attack is taking place. *Ask detection* provides a fix on what the attacker is going after in the *fix* phase, if an attack is indicated. Detecting an attack advances the cycle to the *finish* phase, where response generation is activated.

Each time Panacea successfully elicits a response from the attacker, the new message is *exploited* for attributable information, such as the geographical location of the attack and what organizational affiliations they may have. This information is stored as structured intelligence in the knowledge base which triggers the *analysis* phase, wherein the threat is re-analyzed in a broader context. Finally, Panacea disseminates threat intelligence so that humans can build additional tools and capabilities to combat future threats.

4 Under the Hood

Panacea's main components are presented: (1) Message Analysis Component; and (2) Dialogue Component. The resulting system is capable of handling the thousands of messages a day that would be expected in a modern organization, including failure recovery and scheduling jobs for the future. Figure 2 shows Panacea throughput while operating over a one month backlog of emails, SMS texts, and LinkedIn messages.

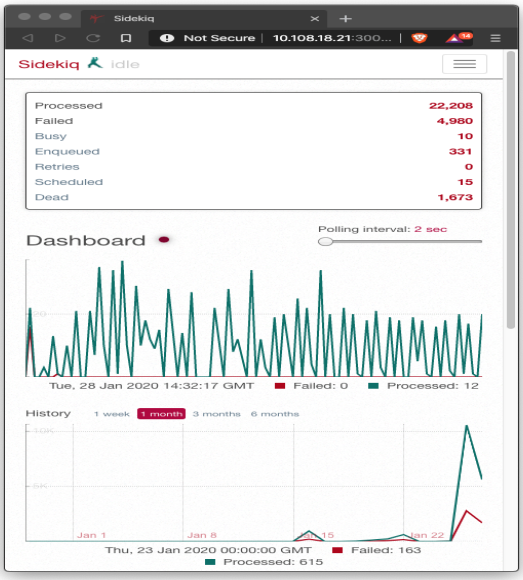

Figure 2: Panacea components run asynchronously in the background for scaling and so new components can be added and removed based on the underlying task.

4.1 Message Analysis Component

Below we describe the structural aspects of messages and their associated processing.

4.1.1 Email Header Classification

When communication takes place over a network, metadata is extracted that serves as a user fingerprint and a source for reputation scoring. Email headers, for example, contain authentication details and information about the mail servers that send, receive, and relay messages as they move from outbox to inbox. To distinguish between benign and malicious emails, Panacea applies a multistage email spoofing, spamming, and phishing detector consisting of: (1) a signature-based detector, (2) an active investigation detector, (3) a receiver-oriented anomaly detector, and (4) a sender-oriented anomaly detector.

4.1.2 Email Content Classification

Dissecting email headers is not enough for detecting malicious messages. Many suspicious elements are related to email bodies that contain user messages related to a specific topic and domain. Analyzing email content provides valuable insight for detecting threats in conversations and a solid understanding of the content itself. Panacea incorporates machine learning algorithms that, alongside of header classifiers, digest email exchanges:

Benign/non-benign classifier: Word embedding vectors (Bengio et al., 2006; Mikolov et al., 2013) trained on email samples from different companies (e.g., Enron) are extracted using neural networks (Sherstinsky, 2013), i.e., back-propagation model with average word vectors as features. This classifier provides a binary prediction regarding the nature of emails (friend or foe).

Email threat type classifier: Spam, phishing, malware, social-engineering and propaganda are detected, providing fine-grained information about the content of emails and support for motive detection (i.e., attacker's intention).

Email zone classifier: Greetings, body, and signature are extracted using word embedding implemented as recurrent neural network with handcrafted rules, thus yielding senders, receivers and relevant entities to enable response generation.

All classifiers support active detection of malicious emails and help in the engagement process of automated bots. Additionally, all trained models have an overall accuracy of 90% using a cross validation approach against well known email collections like Enron (Klimt and Yang, 2004) and APWG (Oest et al., 2018) among other non-public datasets, which makes them reasonably reliable in the context of passive defenses.

4.1.3 Behavioral Modeling

If an adversary is able to compromise a legitimate account, then the header and content classifiers will not be sufficient to detect an attack. The social engineer is able to extract contacts of the account owner and send malicious content on their behalf, taking advantage of the reputation and social relationships attributed to the hijacked account. Two distinctive approaches address these issues:

Impersonation Detector: Sender entities are extracted from the email message and a personalized profile is created for each one, with communication habits, stylometric features, and social network. The unique profiled model is used to assess whether this email has been written and sent by an account's legitimate owner. If a message arrives from a sender that does not have a profile, Panacea applies similarity measures to find other email addresses for the unknown entity. This serves as a defense against impersonation attacks where the social engineer creates an email account using a name and address similar to the user of an institutional account for which a model is available. If Panacea links the unknown account to an institutional account, then that account's model is used to determine whether a legitimate actor is using an unknown account, or a nefarious actor is attempting to masquerade as an insider in order to take advantage of the access such an account would have.

Receiving Behavior Classifier. Individual profiles are built for the receiving behavior of each entity (how and with whom this entity communicates) and new emails are evaluated against the constructed models. To build unique profiles, all messages sent to each particular entity are collected.

4.1.4 Deciders

Panacea must have high confidence in determining that a message is coming from an attacker before deploying active defense mechanisms. A strategy-pattern approach fits different meta-classifiers to different situations. Four classification strategies, called *Deciders*, combine all component analyses after a message is delivered to an inbox to make the final *friend/foe* determination. The Decider API expects all component analyses to include a *friend/foe* credibility score using six levels defined by the Admiralty Code (JDP 2-00, 2011). Deciders may be deterministic through the application of rule based decision making strategies or they may be trained to learn to identify threats based on historical data.

4.1.5 Threat Intelligence

Panacea stores component analysis results in a threat intelligence knowledge base for aggregation of attack campaigns with multiple turns, targets, and threads. The knowledge base adheres to STIX 2.0 specifications and implements MITRE's ATT&CK framework (Strom et al., 2017) to enable attribution and anticipatory mitigations of sophisticated SE attacks. Panacea recognizes indicators of compromise based on features of individual emails as well as historical behavior of senders and recipients. Intrusion sets and campaigns are thus constructed when malicious messages are discovered subsequently linked to threat actors based on attribution patterns, such as IP address, message templates, socio-behavioral indicators, and linguistic signatures. This feature set was prioritized to work with Unit 42's ATT&CK Playbook Viewer. The knowledge base uses a PostgreSQL database backend with an application layer built with Ruby on Rails.

4.2 Dialogue Component

Panacea's dialogue component consists of three key sub-components: *Ask/Framing Detection* (to determine the attacker's demand), *Motive Detection* (to determine the attacker's goal), and **Response Generation** (to reply to suspicious messages).

4.2.1 Ask/Framing Detection

Once an email is processed as described above, linguistic knowledge and structural knowledge are used to extract candidate Ask/Framing pairs and to provide the final confidence-ranked output.

Application of Linguistic Knowledge: Linguistic knowledge is employed to detect both the *ask* (e.g., buy gift card) and the *framing* (e.g., lose your job, get a 20% discount). An ask may be, for example, a request for something (GIVE) or an action (PERFORM). On the other hand, framing may be a reward (GAIN) or a risk (LOSE), for example. Ask/framing detection relies on Stanford CoreNLP constituency parses and dependency trees (Manning et al., 2014b), coupled with *semantic role labeling* (SRL) (Gardner et al., 2017), to identify the main action and arguments. For example, *click here* yields *click* as the *ask* and its argument *here*.

Additional constraints are imposed through the use of a lexicon based on Lexical Conceptual Structure (LCS) (Dorr and Olsen, 2018; Dorr and Voss, 2018), derived from a pool of team members' collected suspected scam/impersonation emails. Verbs from these emails were grouped as follows:

- PERFORM: connect, copy, refer
- GIVE: administer, contribute, donate
- LOSE: deny, forget, surrender
- GAIN: accept, earn, grab, win

Additional linguistic processing includes: (1) categorial variation (Habash and Dorr, 2003) to map between different parts of speech, e.g., *reference*(N) → *refer*(V) enables detection of an explicit ask from *you can reference your gift card*; and (2) verbal processing to eliminate spurious asks containing verb forms such as *sent* or *signing* in *sent you this email because you are signing up*.

Application of Structural Knowledge: Beyond meta-data processing described previously, the email body is further pre-processed before linguistic elements are analyzed. Lines are split where `div`, `p`, `br`, or `ul` tags are encountered. Placeholders are inserted for hyperlinks. Image tags are replaced with their alt text. All styling, scripting, quoting, replying, and signature are removed.

Social engineers employ different link positionings to present "click bait," e.g., "Click here" or "Contact me (jw11@example.com)." Basic link processing assigns the link to the appropriate ask (e.g., *click here*). Advanced link processing ties together an email address with its corresponding PERFORM ask (e.g., *contact me*), even if separated by intervening material.

Confidence Score and Top Ask: Confidence scores are heuristically assigned: (1) Past tense events are assigned low or 0 confidence; (2) The vast majority of asks associated with URLs (e.g., jw11@example.com) are found to be PERFORM asks with highest confidence (0.9); (3) a GIVE ask combined with any ask category (e.g., *contribute $50*) is less frequently found to be an ask, thus assigned slightly lower confidence (0.75); and (4) GIVE by itself is even less likely found to be an ask, thus assigned a confidence of 0.6 (e.g., *donate often*). Top ask selection then selects highest confidence asks at the aggregate level of a single email. This is crucial for downstream processing, i.e., response generation in the dialogue component. For example, the ask "PERFORM contact (jw11@example.com)" is returned as the top ask for "*Contact me. (jw11@example.com).*"

4.2.2 Motive Detection

In addition to the use of distinct tools for detecting linguistic knowledge, Panacea extracts the attacker's intention, or *motive*. Leveraging the attacker's demands (asks), goals (framings) and message attack types (from the threat type classifier), the Motive Detection module maps to a range of possible motive labels: *financial information*, *acquire personal information*, *install malware*, *annoy recipient*, etc. Motive detection maps to such labels from top asks/framings and their corresponding threat types. Examples are shown here:

$$\underbrace{\textit{Give}}_{\text{Ask}} + \underbrace{\textit{Finance info}}_{\text{Ask type}} + \underbrace{\textit{Spam}}_{\text{Email threat}} \rightarrow \textit{Financial info}$$

$$\underbrace{\textit{Gain}}_{\text{Framing}} + \underbrace{\textit{Credentials}}_{\text{Ask type}} + \underbrace{\textit{Malware}}_{\text{Email threat}} \rightarrow \textit{Install malware}$$

These motives are used later for enhancing a response generation process which ultimately creates automatic replies for all malicious messages detected in the Panacea platform.

4.2.3 Response Generation

Response generation is undertaken by a bot using templatic approach to yield appropriate responses based on a hierarchical attack ontological structure and ask/framing components. The hierarchical ontology contains 13 major categories (e.g., *financial details*). Responses focus on wasting the attacker's time or trying to gain information from the attacker while moving along *F3EAD* threat intelligence cycle (Gomez, 2011) to ensure that the attacker is kept engaged. The response generation focuses on the *find, finish and exploit* states. The bot goes after name, organization, location, social media handles, financial information, and is also capable of sending out malicious links that obtain pieces of information about the attacker's computer.

A dialogue state manager decides between time wasting and information seeking based on motive, ontological structure and associated ask/framing of the message. For example, if an attack message has motive *financial details* and ontological structure of *bank information*, coupled with a PERFORM ask, the dialogue state manager moves into an information gathering phase and produces this response: "*Can you give me the banking information for transferring money? I would need the bank name, account number and the routing information. This would enable me to act swiftly.*" On the other hand if the attacker is still after financial information but not a particular piece of information, the bot wastes time, keeping the attacker in the loop.

5 Evaluation

Friend/foe detection (Message Analysis) and response generation (Dialogue) are evaluated for effectiveness of Panacea as an effective intermediary between attackers and potential victims.

5.1 Message Analysis Module

The DARPA ASED program evaluation tests header and content modules against messages for friend/foe determination. Multiple sub-evaluations check system accuracy in distinguishing malicious messages from benign ones, reducing the false alarm rate, and transmitting appropriate messages to dialogue components for further analysis. Evaluated components yield $\sim$90% accuracy. Components adapted for detecting borderline exchanges (*unknown* cases) are shown to help dialogue components request more information for potentially malicious messages.

5.2 Dialogue Module

The ASED program evaluation also tests the dialogue component. Independent evaluators communicate with the system without knowledge of whether they are interacting with humans or bots. Their task is to engage in a dialogue for as many turns as necessary. Panacea bots are able to sustain conversations for an average of 5 turns (across 15 distinct threads). Scoring applied by independent evaluators yield a rating of 1.9 for their ability to display human-like communication (on a scale of 1–3; 1=bot, 3=human). This score is the highest amongst all other competing approaches (four other teams) in this independent program evaluation.

6 Conclusions and Future Work

Panacea is an operational system that processes communication data into actionable intelligence and provides active defense capabilities to combat SE. The F3EAD active defense cycle was chosen because it fits the SE problem domain, but specific phases could be changed to address different problems. For example, a system using the Panacea processing pipeline could ingest academic papers on a disease, process them with components designed to extract biological mechanisms, then engage with paper authors to ask clarifying questions and search for additional literature to review, while populating a knowledge base containing the critical intelligence for the disease of interest.

Going forward, the plan is to improve Panacea's plug-in infrastructure so that it is easier to add capability without updating Panacea itself. This is currently possible as long as new components use the same REST API as existing components. The obvious next step is to formalize Panacea's API. We have found value to leaving it open at this early state of development as we discover new challenges and solutions to problems that emerge in building a large scale system focused on the dangers and opportunities in human language communication.

Acknowledgments

This work was supported by DARPA through AFRL Contract FA8650-18- C-7881 and Army Contract W31P4Q-17-C-0066. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of DARPA, AFRL, Army, or the U.S. Government.

References

Taimur Bakhshi, Maria Papadaki, and Steven Furnell. 2008. A practical assessment of social engineering vulnerabilities. In *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance*.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. *Neural Probabilistic Language Models*, pages 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ram Bhakta and Ian G. Harris. 2015. Semantic analysis of dialogs to detect social engineering attacks. *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 424–427.

Deanna D Caputo, Shari Lawrence Pfleeger, Jesse D Freeman, and M Eric Johnson. 2013. Going spear phishing: Exploring embedded training and awareness. *IEEE Security & Privacy*, 12(1):28–38.

Vitor Carvalho and William Cohen. 2004. Learning to extract signature and reply lines from email. In *First Conference on Email and Anti-Spam*, pages 1–8. CEAS.

Emmanuel G. Dada, Joseph S. Bassi, Haruna Chiroma, Shafi'i M. Abdulhamid, Adebayo O. Adetunmbi, and Opeyemi E. Ajibuwa. 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):1–23.

Dorothy E. Denning. 2014. Framework and principles for active cyber defense. *Computers & Security*, 40:108–113.

Bonnie Dorr and Clare Voss. 2018. STYLUS: A resource for systematically derived language usage. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 57–64, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Bonnie J. Dorr and Mari Broman Olsen. 2018. Lexical conceptual structure of literal and metaphorical spatial language: A case study of push. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 31–40.

Nouha Dziri, Ehsan Kamalloo, Kory W Mathewson, and Osmar Zaiane. 2018. Augmenting neural response generation with context-aware topical attention. *arXiv preprint arXiv:1811.01063*.

Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1065–1074, New York, NY, USA. ACM.

Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational ai. *Foundations and Trends in Information Retrieval*, 13(2-3):127–298.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*.

Mike Goffin. 2020. CRITs - Collaborative Research Into Threats.

Jimmy A. Gomez. 2011. The targeting process: D3A and F3EAD. *Small Wars Journal*, 1:1–17.

Nizar Habash and Bonnie J. Dorr. 2003. A categorial variation database for english. In *In Proceedings of the Human Language Technology and North American Association for Computational Linguistics (NAACL) Conference*, pages 96–102.

Christopher Hadnagy and Michele Fincher. 2015. *Phishing Dark Waters*. Wiley Online Library.

Christopher Hidey and Kathleen McKeown. 2018. Persuasive Influence Detection: The Role of Argument Sequencing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5173–5180, San Francisco, California, USA.

JDP 2-00. 2011. *Understanding and intelligence support to joint operations (JDP 2-00)*. Ministry of Defence (UK).

A. Karakasilitiosis, S. M. Furnell, and M. Papadaki. 2006. Assessing end-user awareness of social engineering and phishing. In *Proceedings of the Australian Information Warfare and Security Conference*.

Myeongsoo Kim, Changheon Song, Hyeji Kim, Deahyun Park, Yeeji Kwon, Eun Namkung, Ian G Harris, and Marcel Carlsson. 2018. Catch me, yes we can!-pwning social engineers using natural language processing techniques in real-time.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226. Springer Berlin Heidelberg.

Andrew Lampert, Robert Dale, and C'ecile Paris. 2009. Segmenting email message text into zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, page 919–928. Association for Computational Linguistics.

Chang Li and Dan Goldwasser. 2019. Encoding social information with graph convolutional networks forPolitical perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2594–2604, Florence, Italy. Association for Computational Linguistics.

Yu Li, Kun Qian, Weiyan Shi, and Zhou Yu. 2020. End-to-end trainable non-collaborative dialog system. *ArXiv*, abs/1911.10742.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014a. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60. ACL.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014b. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119. Curran Associates Inc.

Adam Oest, Yeganeh Safei, Adam Doupe, Gail-Joon Ahn, Brad Wardman, and Gary Warner. 2018. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *Proceedings of the 2018 APWG Symposium on Electronic Crime Research, eCrime 2018*, pages 1–12. IEEE Computer Society.

Daniela Oliveira, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin, and Natalie Ebner. 2017. Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 6412–6424, New York, NY, USA. ACM.

Tiancheng Zhao Amy Pavel Maxine Eskenazi Prakhar Gupta, Shikib Mehri and Jeffrey Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proceedings of the 20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Stockholm, Sweden. Association for Computational Linguistics.

Abhilasha Ravichander and Alan W. Black. 2018. An empirical study of self-disclosure in spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, pages 253–263.

Sashank Santhanam and Samira Shaikh. 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. *arXiv preprint arXiv:1906.00500*.

Yuki Sawa, Ram Bhakta, Ian Harris, and Christopher Hadnagy. 2016. Detection of social engineering attacks through natural language processing of conversations. In *Proceedings of the 2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 262–265.

Alex Sherstinsky. 2013. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. pages 1–39. Cornell University.

Blake E. Strom, Joseph A. Battaglia, Michael S. Kemmerer, William Kupersanin, Douglas P. Miller, Craig Wampler, Sean M. Whitley, and Ross D. Wolf. 2017. Finding cyber threats with att&ck-based analytics. *The MITRE Corporation, Tech. Rep.*, 1(1).

Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. Misp: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, pages 49–56. ACM.

Meijuan Yin, Xiao Li, Junyong Luo, Xiaonan Liu, and Yongxing Tan. 2011. Automatically extracting name alias of user from email. *International Journal of Engineering and Manufacturing*, 1:14–24.

Yue Zhang, Serge Egelman, Lorrie Faith Cranor, and Jason I. Hong. 2006. Phinding phish: Evaluating anti-phishing tools. Technical report, Carnegie Mellon University.