Computers in Human Behavior 80 (2018) 207–215

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

What it takes to get retweeted: An analysis of software vulnerability messages

Romilla Syed*, Maryam Rahafrooz, Jeffrey M. Keisler

College of Management, University of Massachusetts, Boston, 100 Morrissey Blvd., Boston, MA 02125-3393, USA

ARTICLE INFO

Article history:
Received 9 August 2017
Received in revised form
16 November 2017
Accepted 17 November 2017
Available online 20 November 2017

Keywords:
Software vulnerability
Hazard messages
Twitter
Terse communication
Information diffusion

ABSTRACT

A large body of research has examined the public disclosure of software vulnerability, but little attention has been paid to sharing software vulnerability information on social media. Sharing software vulnerability messages on Twitter indicates that particular messages are perceived by the public valuable enough to share with others. Building on hazard communication and terse messaging literature, this study analyzes the factors impacting the retweeting of software vulnerability related messages. Particularly, this study has two goals: 1) to identify the major content categories contained in software vulnerability related tweets and 2) to understand the impact of tweet content, tweet source, technical features of tweets, as well as software vulnerability features on retweeting the software vulnerability messages. Our analysis suggested five content categories are referred in the tweets: alerts, patch, advisory, exploit, and root-cause. Using a negative binomial regression, we found that several factors jointly influence the retweeting of software vulnerability messages. The findings could be useful for planning about effective message design for communicating the publicly disclosed software vulnerability information to end-users.

Published by Elsevier Ltd.

1. Introduction

Recent years have seen an increased use of social media for discussing and sharing information about natural and technical hazards (Sutton et al., 2015a, 2015b; Syed & Dhillon, 2015). The United States Computer Emergency Response Team Coordination Center (US-CERT/CC) has also adopted social media channels for sharing latest software vulnerability alerts and advisories.[1] The US-CERT is a cybersecurity division of the Department of Homeland Security created in 2003. It is responsible for analyzing and reducing cyber threats and vulnerabilities, disseminating cybersecurity information to the public, and coordinating computer security incident response activities. A software vulnerability is a weakness or flaw in a system that if exploited could threaten the confidentiality, integrity, or availability of the system (Mell, Scarfone, & Romanosky, 2007). Recent surveys report that Twitter, a popular social media site, is increasingly used to share software vulnerability information (NopSec, 2015). The diffusion of software vulnerability information on Twitter could be helpful to inform software vendors, developers, and end users about the vulnerabilities or patch availability. While a growing body of research has explored the key mechanisms through which information diffuses on social media (e.g., Chung, 2017; Pang & Law, 2017; Sutton et al., 2015a), there is a lack of understanding about the role of social media in diffusing software vulnerability information.

Prior research analyzes the behavioral effects of receiving terse messages on social media during imminent threat or hazard situations (Sutton et al., 2015b). Terse messages are defined “as brief messages that are easily shared and quickly propagated, have the potential to reach large numbers of online users, in real time, disseminating information at critical points of a hazard event” (Sutton et al., 2015b, p. 20). Terse messages are intended to provide situational awareness and to instruct the public at risk about the protective actions. In this study, we extend the concept of terse messaging to analyze the retweeting of software vulnerability information on Twitter. Retweeting refers to the act of passing on Twitter messages to others that one has received from some third party. Building on hazard literature, we conceptualize software vulnerability as a technical hazard to information resources, systems, and networks (Mell et al., 2007; Sorensen, 2000). We further

* Corresponding author.
E-mail addresses: Romilla.Syed@umb.edu (R. Syed), Maryam.Rahafrooz001@umb.edu (M. Rahafrooz), Jeff.Keisler@umb.edu (J.M. Keisler).

[1] https://www.us-cert.gov/.

https://doi.org/10.1016/j.chb.2017.11.024
0747-5632/Published by Elsevier Ltd.

argue that an increased volume of retweeting indicates that the vulnerability information is actively attended to by social media members. Retweeting software vulnerability messages is also indicative of the fact that certain features of messages are perceived to be of some intrinsic value by the public (see Sutton et al., 2015a). Hence, we focus on multiple features of software vulnerability messages including message content, message source, key technical features of messages as well as the features of software vulnerabilities to answer the following two research questions.

RQ1. What are the major content categories referred in the software vulnerability related terse messages?

RQ2. What are the features of software vulnerability related terse messages that predict their retweetability on Twitter?

The rest of the paper is organized as follows. Section 2 presents a review of prior research related to software vulnerabilities and hazard communication on social media. We also review the factors impacting message retweeting. Section 3 discusses our approach for data collection, processing, and analysis. Section 4 summarizes the results. Finally, we discuss the implications of our research in Section 5. The limitations of this research and future research directions are also noted.

2. Background literature

In this section, we first review the existing research related to software vulnerability management. Next, we discuss the theoretical rationale of terse communication during hazard situations. Finally, we review the literature related to information retransmission on social media.

2.1. Software vulnerability management

In recent years, a growing body of scholarly work has begun to study software vulnerabilities. Overall four distinct streams of literature have emerged. The first stream relates to the *implications of vulnerability disclosure*. In one of the earliest studies, Telang and Wattal (2005) analyzed the negative impact of vulnerability disclosure on the market value of a software vendor. Several other studies have examined the influence of vulnerability disclosure on attack volume or frequency. For example, Arora, Nandkumar, and Telang (2006) analyzed the frequency of attacks conditioned by vulnerability disclosure. The authors note that unpatched vulnerabilities attract fewer attacks than patched vulnerabilities irrespective of whether the vulnerability is secret or publicly disclosed. In another study, Ransbotham and Mitra (2013) examined the impact of immediate vulnerability disclosure by security professionals on attack diffusion and volume. Immediate disclosure accelerates dissemination and penetration of attacks and risk of attack but decreases the volume of attack. In a related study, Ransbotham, Mitra, and Ramsey (2012) found that market disclosure (i.e., markets that reward for vulnerability discovery) delays the diffusion of attacks, reduces the risk of the first attack, and reduces the volume of attacks. Finally, Wang, Xiao, and Rao (2010) studied the impact of network attacks and vulnerability disclosure on Internet users' search behavior. This study shows that the network attacks on the current day and the day before affect the search behavior whereas vulnerability disclosure does not.

The second stream of literature relates to *vendor response* to vulnerability disclosure. Several studies have analyzed the impact of vulnerability disclosure on vendors patch release behavior. For example, Arora, Krishnan, Telang, and Yang (2010) found that vulnerability disclosure accelerates vendors' patch release timing. However, vendors release patches slowly for vulnerabilities disclosed by private parties such as SecurityFocus in comparison to those disclosed by CERT/CC. As expected, vendors are reported to be more responsive to more severe vulnerabilities. Further, open source vendors release patches more quickly. Temizkan, Kumar, Park, and Subramaniam (2012) further explored the effect of vulnerability disclosure on patch release timing. Unlike previous studies (e.g., Arora et al., 2010) who treat vulnerability severity as an aggregate measure, Temizkan et al. (2012) empirically tested the impact of confidentiality, integrity, and availability on vendors patch release behavior.

The third stream of literature relates to the *vulnerability disclosure policy*. In one of the earliest studies, Li and Rao (2007) examined the effect of private intermediaries such as iDefense and TippingPoint on the disclosure process. They show that private intermediaries' participation in the vulnerability disclosure does not affect the optimal disclosure timing of the public intermediary, i.e., CERT/CC. In a related study, Arora, Telang, and Xu (2008) proposed a framework to analyze the optional timing of vulnerability disclosure. The authors found that a longer protected period (i.e., withholding public disclosure) does not increase the patch quality.

Finally, the fourth stream of literature explores the *role of social media in software vulnerability disclosure*. Particularly, these studies focus on the usefulness of social media sites such as Twitter for drawing vulnerability intelligence. For example, Trabelsi et al. (2015) note that several communication media are used to issue security advisories about vulnerabilities, workarounds, and patches. However, it is time-consuming to gather intelligence from multiple heterogeneous sources to get a comprehensive view of vulnerabilities. The authors proposed a software vulnerability monitoring systems based on the security information collected from Twitter. In a related study, Mittal, Das, Mulwad, Joshi, and Finin (2016) proposed an ontology-based system to generate early alerts for cybersecurity threats and vulnerabilities. The use of Twitter feeds for detecting exploits, calculating risk, and prioritizing response actions has also been noted (Joh & Malaiya, 2010; Sabottke et al., 2015).

In summary, the existing research has increased our understanding of the vulnerability disclosure and risk management. There is also some focus on integrating social media intelligence for software vulnerability management. However, there is a lack of empirical research focusing on the diffusion of software vulnerability information on social media. As noted before, a higher volume of retweets indicates that software vulnerability information is actively attended to by Twitter users. Building on hazard literature, our study explores the factors that lead to retweeting of software vulnerability information.

2.2. Hazard communication

Warning messages are issued to the public in response to an imminent threat posed by hazard situations such as natural disasters (e.g., earthquakes, volcanic eruptions, hurricanes, and floods), technological disasters (e.g., nuclear power and chemical transportation), and national security events (e.g., nuclear attacks and terrorist activities) (Sorensen, 2000; Sutton et al., 2015a, 2015c). The messages are intended to inform and instruct the public at risk about the severity of the hazard and precautionary measures for preventing the harm. This body of literature has built on the theories of collective behavior and emergent norms (Blumer, 1951; Turner, 1987) and has focused on the effects of message channels, sources, content, and hazard type on the behavioral responses of individuals (Drabek, 1999; Mayhorn & McLaughlin, 2014; Sutton et al., 2015b). Several studies also note the effect of personal, social, and situational factors on an individual's behavioral response (Lindell, 1987).