THE RELATIONSHIP BETWEEN THE CHARACTERISTICS OF SOFTWARE DEVELOPERS AND SECURITY BEHAVIOR INTENTION

Wisdom Umeugo

Independent Researcher, Ottawa, Canada

ABSTRACT

Software security is integral to information security, requiring software developers to be aware of software security as implementers and information security as employees. Little is known about the impact of software developers' characteristics on their information security behaviors. This study examined the relationship between software developer characteristics such as their application security awareness (AW), training, and self-efficacy (SE) and their information security behavior intention (SeBI). Data from a survey of 200 software developers in the United States were analyzed using correlational statistical methods. AW and SE positively correlated with SeBI, while application security training did not correlate. Developers' AW and SeBI mean score was found to be poor. Information security managers should aim to improve software developers' application security awareness and self-efficacy to help improve their information security behavior. Application security training infused with information security awareness and conducted by application security experts is recommended.

KEYWORDS

Self-efficacy, Software security, Information security, Security behavior, Awareness, Application security

1. INTRODUCTION

Software security is a vital part of information security. Software security, like all aspects of information security, guarantees the confidentiality, integrity, and availability of information systems and data [1]. Software is a chief enabler of digital transformation and now controls most domestic and economic activities [2]. Most security incidents result from software vulnerability exploits [3]. Therefore, ensuring software integrity is critical to protecting cyberinfrastructure and reducing cyberattack risks [3]. Researchers have recommended practicing a security-infused secure software development lifecycle (SSDLC) to enhance the security of released software [1], [4]–[6]. Building secure software requires staff with expertise in threat analysis, security engineering, attack surface determination, and security architectures [7]. Software developers generally need to gain these software security skills. However, they are still entrusted with software security in small and medium enterprises (SMEs) due to the high costs of maintaining a skilled software security team or poor management attitudes towards software security [7]. In such situations, developers become critical players in the security of software developed and used by their organization and in organization-wide information security as employees and information systems users. Software developers with security responsibilities are likely to receive application security training as part of their duties, which may impact their knowledge, awareness, self-efficacy, and attitude toward general information security. This study empirically examined the effect of software developers' application security awareness, application security training, and application security self-efficacy on their information security behavior. Existing studies focus on software developers' application security self-efficacy, motivation, and attitude

without considering the impact on their information security behavior. Furthermore, information security behavior studies focus broadly on general populations, such as students and employees [8]. None have been explicitly applied to software developers. This study fills the existing gap in knowledge of the effect of software developers' application security awareness, application security training, and application security self-efficacy on their information security behavior.

2. BACKGROUND

2.1. Security Behavior Intention

Security behavioral intentions are individuals' desire to engage in a given behavior [9]. Behavioral intentions are studied in information systems research as predictors of actual behavior and precursors of planned behavior [10], [11]. Common intended behaviors in published information security studies include protective behaviors, compliance, usage behavior, and adoption. Behavior intention has been shown to correlate with actual behavior positively [10], [12]–[15]. Fishbein and Ajzen's theory of reasoned action (TRA) and Ajzen's theory of planned behavior (TPB) are two popular psychology theories dealing with behavioral intention [16]. Both theories view behavioral intention as a predictor of actual behavior. TRA proposed that behavioral intentions are an immediate precursor of behavior and consist of an attitudinal and normative factor [17], [18]. TRA defines behavioral intentions as an individual's subjective probability of performing a given behavior. TPB extended TRA by considering a third factor called perceived behavioral control (PBC), which represents the resources and opportunities required to complete the given behavior [18]. PBC influences the motivation to perform a given task, directly affecting behavioral intention [18]. Various other factors have been proposed to influence behavioral intentions directly or indirectly, such as performance expectancy, effort expectancy, social influence, autonomy, competence, relatedness [14], self-efficacy, response cost, response efficacy, perceived severity, perceived vulnerability [19], [20], descriptive norms, social support [20], organizational identification, security-related organizational justice [21], relative advantage, compatibility, and complexity [22]. These factors are mostly derived from a conjunction of TPB with the Protection motivation theory and the Diffusion of innovation theory.

2.2. Self-Efficacy

Self-efficacy refers to an individual's belief in their ability to perform a given task [23]. Self-efficacy is a construct from social cognitive theory. Self-efficacy can be explained as a personal judgment of one's belief in one's ability to complete a given task [24]. Self-efficacy is a predictor of motivation and performance [23]. Self-efficacy determines one's choices of tasks, motivation levels, exertion, and perseverance at a given task [25]. When faced with a challenging task, people with low self-efficacy may avoid the task or capitulate due to fear or stress and choose an optional task they feel they can successfully execute [25]. Conversely, a person with high self-efficacy will persevere on a challenging task due to belief in their abilities [26].

According to Bandura [26], there are four sources of self-efficacy: mastery, vicarious experiences, social persuasion, and physiological state. Mastery or performance accomplishments refer to historic success and accomplishments at a given or related task [26]. More success achieved with specific tasks leads to higher beliefs in one's ability and higher self-efficacy, while failures lead to lower self-efficacy [23]. Vicarious experiences are experiences from observation of the execution of tasks by peers [26]. People learn through observation of others and judge how well they can perform at a given task. Vicarious experiences provide a comparison and achievement model that the observer accepts and imitates [27]. Social persuasions are social influences such as encouragement to persuade an individual of their ability to cope and persist

with a task. Social persuasions are typically given verbally during the start or execution of a task. Social persuasion may be provided by friends, family, peers, co-workers, and instructors. Appelbaum and Hare [23] warned against social persuasions with unrealistic expectations because repeated failures may lead to adverse effects on the persuaded individual. Physiological arousal is the emotional state induced by a given task [26]. These emotions may be fear, pain, anxiety, agitation, and stress [23], [26]. Emotional feedback provides information for judging one's ability to complete the task. Negative emotions result in lower self-efficacy perception [23]. In information security behavioral studies, self-efficacy is frequently measured as a factor and a predictor of behavioral intentions and actual behavior. Gao et al. [28] found that self-efficacy accurately predicted behavioral intentions and actual behavior during the behavioral adoption stage in their study. Bulgurcu et al. [29] reported a positive effect of self-efficacy on information security policy compliance intention. In information technology and information systems studies, self-efficacy has been shown to influence usage intentions and innovation adoption [30], [31]. A high information security self-efficacy in employees is essential to information security management. Employees should be motivated to comply with information security policies and have the self-efficacy to take protective actions. A high security self-efficacy is also critical for information security staff tasked with implementing and monitoring security because of the increasing breadth and sophistication of attack methods and security tools.

3. RELATED WORK

There is a wealth of existing studies on the information security behavior of collective groups. Various researchers have studied factors impacting information security awareness and behavior. The population of interest in these studies is usually employees of organizations and employment sectors and students of academic institutions [8]. None have examined software developers' information security behaviors. Existing behavioral Studies focused on software developer populations focus on their application security behaviors. Balebako et al. [32] Interviewed software developers on their privacy and security behaviors. Only a few developers among the interviewees had formal training on privacy and security. Those that had training typically received it through corporate training or industry certification. Balebako et al. [32] found that developers mainly consulted online sources and their social networks for privacy and security advice. Other findings revealed that app developers did not prioritize creating privacy policies because they believed privacy policies entailed legalities that turned off users.

Arizon-Peretz et al. [33] studied the role the organizational work environments of developers play in forming their mindsets and behavior. Their study used organizational climate theory to understand better developers' perceptions and behaviors and the organizational forces affecting them. Qualitative analysis of interviews with 27 software developers from 14 companies showed that software developers are faced with inconsistent and confusing cues by management in their work environment. Developers perceived these cues as low-priority leading to behaviors that did not align with the expectation and recommendations of policymakers [33].

Van der Linden et al. [34] sought to uncover the rationale underpinning developers' application security decisions. Their research involved two studies, a security task-based study and a survey of 274 developers. Results showed that developers rarely rationalize their decisions using security considerations when faced with non-coding tasks. Their results indicated that developers consider security carefully only when coding.

Jing Xie et al. [35] investigated how and why software developers produce security bugs. They conducted a semi-structured interview with 15 software developers to understand their software security perceptions and behaviors. Results showed a disconnect between developers' conceptual understanding of security and their attitudes toward their security responsibilities. Some

developers interviewed said they had knowledge of security issues of various platforms but did not feel it concerned them because their work did not involve those platforms. Other reasons developers used to rationalize their unwillingness to take software security seriously were that very few people used their applications and that their users were incapable of malicious intent.

4. HYPOTHESES DEVELOPMENT

4.1. Application Security Awareness

In information security research, awareness is generally viewed as awareness of information security policies. Awareness is defined as the degree of knowledge of information security policies and the behavior toward complying with information security policies [36], [37]. Awareness is often expressed in three dimensions: knowledge, attitude, and behavior [38]. Similarly, application security awareness (AW) is defined as the knowledge of application security policies, guidelines, and best practices and the compliance behavior with these policies, procedures, and best practices. Security behavior intention has been shown to correlate positively with information security awareness [39], [40]. Higher application security awareness is likely to instill broader security consciousness and positively impact security behaviors. Developers that emphasize and implement security are also expected to accept and practice security behaviors such as strong passwords and information privacy that they implement in software. Therefore, it is posited that application security awareness positively correlates with security behavior intention.

H1: Application security awareness positively correlates with information security behavior intention.

4.2. Application Security Training

Application security training (TR) involves training software developers to conduct application security tasks. Such security tasks include risk analysis, secure coding, secure software testing, and security engineering [7], [41]. Training on any facet shows employees that the management views that facet as necessary [33]. Therefore, application security training shows that management values application security. Application security training improves application security awareness and instills security values in software developers. Application security training also provides software developers perspectives on insecure practices, the mechanism of security compromise through software vulnerability exploits, implications of successful attacks, and general protective security habits. For this reason, it is posited that application security training positively correlates with information security behavior intention.

H2: Application security training positively correlates with information security behavior intention.

4.3. Application Security Self-efficacy

Application security Self-efficacy (SE) is one's self-belief in one's ability to execute application security tasks successfully [42]. Self-efficacy has a generality dimension that refers to how applicable self-efficacy in a given task is to related tasks [23]. Perceptions of high self-efficacy tend to generalize to related tasks [26]. Application security self-efficacy may generalize to information security self-efficacy, which has been shown to have a positive correlation with information security behavior [43]–[45]. Therefore, a positive relationship is posited between application security self-efficacy and information security behavior intention.

H3: Application security self-efficacy positively correlates with information security behavior intention.

This study examines two other factors affecting software developers' application security self-efficacy: programming language and software deployment platform. Software deployment platforms are the platforms where software is deployed. Deployment platforms include cloud, web and Internet, industrial systems, mainframe, mobile devices, personal computers, smart devices, and the Internet of Things (IoT). The programming language and the platform deployed determine the prevalence and types of security vulnerabilities software developers deal with. Various memory-related weaknesses and some vulnerabilities listed in vulnerability databases, such as the Common Weakness Enumeration and Open Web Application Security Project (OWASP) top 10, are tied to programming languages and application platforms. The inherent security features and architecture built into programming languages and deployment platforms may influence developers' application security self-efficacy. Developer application security self-efficacy may also be affected by the amount of available security information, the availability of published security guidelines and best practices, and the size of the developer community of both the programming language used and the deployment platform. Developers mainly consult online sources and their social networks for privacy and security advice [32]. Therefore, software developers' ease of finding security information and solutions related to their programming language and deployment platform impacts their application security self-efficacy. Programming languages and deployment platforms vary in the amount of available security information and community activity. It is therefore posited that developer application security self-efficacy differs by the programming language and deployment platform.

H4: Application security self-efficacy statistically significantly differs by the programming language.

H5: Application security self-efficacy statistically significantly differs by the deployment platform.

5. RESEARCH METHODOLOGY

The study was quantitative non-experimental correlational. An online survey questionnaire was used as the data collection instrument. Data collected was analyzed using Spearman's correlation and one-way ANOVA. The questionnaire used was hosted online on Pollfish. The target population was software developers working full-time in organizations based in the United States. A random sample from the target population was recruited through Pollfish's audience service.

The questionnaire was close-ended, with demographic questions and Likert-scale questions measuring the study's variables. The demographics questions in the survey requested participants' sex, age range, primary programming language, application security training, and application deployment platform. Participants' application security self-efficacy (SE) was measured using the 15-item validated secure software-development self-efficacy scale (SSD-SES) published by [42]. Participants' self-reported information security behavior intention (SeBI) was measured using the SeBI scale (SeBIS) created by [10]. SeBIS consists of 16 questions measured using a five-point Likert-scale measure Never (1), Rarely (2), Sometimes (3), Often (4), and Always (5).

Social desirability bias (SDB) was evaluated because the SE and SeBI measures were self-reported. SDB occurs when respondents choose socially desirable responses by over-reporting or under-reporting their behavior [46]. SDB is undesirable and contaminates measured variables. The effect of SDB can be evaluated by estimating the degree of correlation between the SDB

measure and other behavioral variables [47]. SDB is generally ignored if there is no significant correlation or the correlation size is trivial [47]. SDB was measured in the questionnaire using the five items from the Socially Desirable Response Scale (SDRS-5) proposed by Hays et al. [48]. The correlation between SDB, SE, and SeBI was estimated to determine the effect of SDB on the study. Participants were also informed of the anonymous nature of the study to reduce SDB.

A total of 200 valid responses were accepted from 290 responses after filtering out ineligible responses and removing incompletes and speeders. The data was imported into Jamovi statistical software for analysis. Descriptive statistical analysis was performed to examine the study's demographics. Participants' SE and SeBI scores were averaged to obtain a mean score of 5 or less. The interpretation of the SE score was based on [49]. A score between 4.5 and 5 was high, 4.0 – 4.5 average, and scores lower than 4.0 were poor. Spearman's Rho inter-variable correlation was performed to determine correlations. Spearman's Rho correlation coefficient measures the strength of the association between variables with a value between -1 and +1. The sign of the Spearman rho coefficient indicates the direction of the association. The statistical significance of the association was determined using the Spearman rho p-value (p). A statistically significant relationship is characterized by a p-value less than 0.05 ($p < .05$). Non-parametric ANOVA or Kruskal-Wallis test was used to determine if there were significant differences in SE scores between programming language and deployment platform groups.

6. RESULTS

A total of 269 respondents participated in the survey. Two hundred responses were accepted after screening and data quality checking. One hundred and two or 51% of the participants were female, and 98 (49%) were males. Most participants were aged between 35 – 44 (n=82). JavaScript and Java were the most cited primary programming languages, accounting for 20% and 17.5% of participants' primary programming languages. C++ (n=31), Python (n=26), and PHP (n=18) were the following popular languages making up 15%, 13%, and 9% of participants' primary programming languages, respectively. Cloud, web, and internet (n=59) was the most deployed application platform among respondents. Mobile devices (n =53), smart devices (n=38), and personal computers (n=21) were the following most stated primary deployment platforms. One hundred and twenty-seven participants claimed they were not given application security training, while 73 claimed they had application security training. Table 1 summarizes the participant demographics.

Table 1. Participant demographics

Demographic variable	Group	Counts	% of Total
Gender	female	102	51.0 %
	male	98	49.0 %
Age	18 - 24	23	11.5 %
	25 - 34	73	36.5 %
	35 - 44	82	41.0 %
	45 - 54	16	8.0 %
	> 54	6	3.0 %
Programming Language	C	6	3.0 %
	C#	6	3.0 %
	C++	31	15.5 %

	Go	10	5.0 %
	Java	35	17.5 %
	Javascript	40	20.0 %
	Kotlin	8	4.0 %
	PHP	18	9.0 %
	Perl	2	1.0 %
	Python	26	13.0 %
	Ruby	2	1.0 %
	Scala	3	1.5 %
	Swift	7	3.5 %
	Typescript	6	3.0 %
Deployment Platform	Cloud, Web, and Internet	59	29.5 %
	Industrial systems	19	9.5 %
	Mainframe	10	5.0 %
	Mobile devices	53	26.5 %
	Personal computers	21	10.5 %
	Smart devices and the Internet of Things (IoT)	38	19.0 %
Application Security Training	No	127	63.5%
	Yes	73	36.5%

The scales for awareness, self-efficacy, and SeBI were all tested for reliability using Cronbach's alpha. All scales had good Cronbach alpha scores above 0.6. Table 2 summarizes the Cronbach alpha scores of the measurement scales.

Table 2. Cronbach's alpha for the variables

Variable	**Cronbach's α**
Awareness	0.703
Self-Efficacy	0.939
SeBI	0.662

The Inter-variable Correlation Matrix was calculated from the data. Table 3 shows the inter-item correlation matrix. The effect of social desirability bias (SDB) was evaluated by examining Spearman's rho for the correlation of all items with the SDB variable. Self-efficacy ($r_s(298) = .143$, $p < .05$) and SeBI ($r_s(298) = .211$, $p < .01$) statistically significantly correlated with SDB. However, the correlation coefficients were less than 0.3, a weak correlation [50]. There was no statistically significant correlation between SDB and awareness. Therefore, the effects of SDB were ignored in the study. SE and SEBI scores for all participants were calculated. The SE mean score was 3.82, while the SeBI mean score was 3.45, both poor scores.

Table 3. Inter-variable correlation matrix

		SDB	**SE**	**AW**	**SEBI**
SDB	Spearman's rho	—			

	p-value	—			
SE	Spearman's rho	-0.143*	—		
	p-value	0.044	—		
AW	Spearman's rho	-0.055	0.542***	—	
	p-value	0.438	< .001	—	
SEBI	Spearman's rho	0.211**	0.300***	0.274***	—
	p-value	0.003	< .001	< .001	—

6.1. Hypothesis One

Application security awareness positively correlated with SeBI ($r_s(298) = .274, p < .001$). Hypothesis One, stating that application security awareness positively correlates with information security behavior intention, was, therefore, supported.

6.2. Hypothesis Two

Kendall's tau-b correlation was calculated to determine the relationship between application security training and SeBI. There was a weak negative association between application security training and SeBI, which was not statistically significant ($\tau b = -.043, p = .466$). Hypothesis Two, stating that application security training positively correlates with information security behavior intention, was, therefore, unsupported.

6.3. Hypothesis Three

Application security self-efficacy positively correlated with SeBI ($r_s(298) = .3, p < .001$). Hypothesis Three, stating that application security self-efficacy positively correlates with information security behavior intention, was supported.

6.4. Hypothesis Four

Table 4 shows each programming language group's developer application security self-efficacy descriptives. Ruby and C# languages had the highest self-reported application security self-efficacy, with mean application security self-efficacy scores of 4.4 and 4.33, respectively. However, Ruby and C# needed to be more represented in the sample, having just 2 and 6 adopted developers, respectively. The popular languages, Java and Javascript, had reported application security self-efficacy mean scores of 3.92 and 3.63. Developers using Perl had the lowest self-reported application security self-efficacy score of 2.97. Figure 1 graphical depicts the self-efficacy scores by programming language.

Table 4. Programming languages self-efficacy descriptive

Programming language	**Frequency (N)**	**Self-efficacy Mean score**
C	6	3.66
C#	6	4.33
C++	31	3.87
Go	10	3.92

Java	35	3.92
Javascript	40	3.63
Kotlin	8	3.8
PHP	18	3.92
Perl	2	2.97
Python	26	3.69
Ruby	2	4.4
Scala	3	3.38
Swift	7	4.1
Typescript	6	3.87

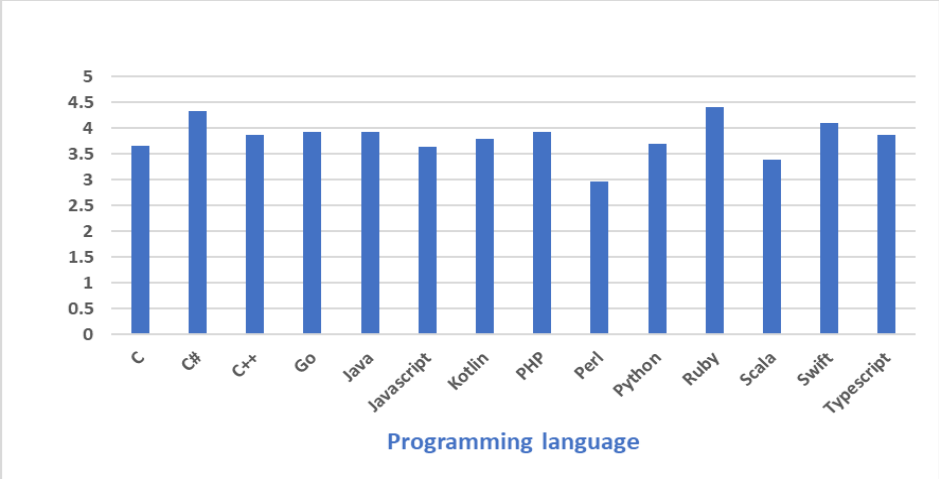

Figure 1. Self-efficacy scores by programming language

A one-way ANOVA test was used to determine if there were any differences in developer self-efficacy between groups of primary programming languages. However, the data failed the ANOVA normality test requirement with a statistically significant Shapiro-Wilk test ($W = 0.95$, $p < 0.001$). The non-parametric one-way ANOVA or the Kruskal-Wallis test was used to determine if the differences in developer application security self-efficacy between groups of programming languages were statistically significant. Results showed that there was no statistically significant difference in developer's application security self-efficacy between programming language groups ($\chi 2$ (13) = 15.6, $p > .05$). Hypothesis Four, stating that application security self-efficacy statistically significantly differs by programming language, was therefore, unsupported.

6.5. Hypothesis Five

Table 5 shows the developer application security self-efficacy descriptives for each deployment platform group. Developers that deployed applications for industrial systems and IoT had the highest reported mean application security self-efficacy score of 4.06. Developers who deployed to the cloud, web, and internet (CWI) and personal computers had the lowest reported application security self-efficacy, with mean scores of 3.41 and 3.67, respectively. Figure 2 graphically depicts the self-efficacy scores by deployment platform.

Table 5. Deployment platform self-efficacy descriptive

Platform	Frequency (N)	Self-efficacy score
Cloud, Web, and Internet	59	3.41
Industrial systems	19	4.06
Mainframe	10	4.04
Mobile devices (Phones, PDAs, and Smart watches)	53	4.03
Personal computers	21	3.67
Smart devices and the Internet of Things (IoT)	38	4.06

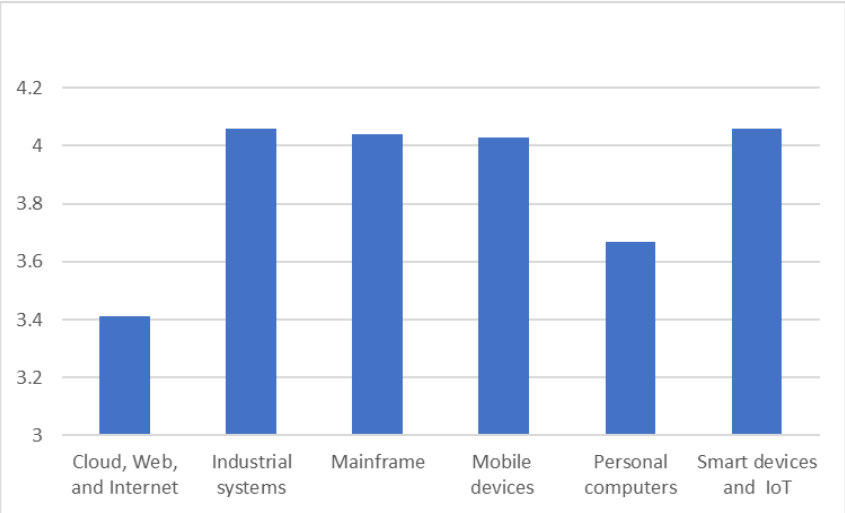

Figure 2. Self-efficacy scores by deployment platform

A one-way ANOVA test was used to determine if there were any differences in developer self-efficacy between groups of deployment platforms. The data failed the normality test requirement with a statistically significant Shapiro-Wilk test ($W = 0.97$, $p < 0.001$). Non-parametric one-way ANOVA was used to determine if the differences in developer application security self-efficacy between groups of deployment platforms were statistically significant. Results showed that there was a statistically significant difference in the developer's application security self-efficacy with the deployment platform ($\chi 2$ (5) = 25.5, $p < .001$). Hypothesis Five, stating that application security self-efficacy statistically significantly differs by the deployment platform, was supported. Dwass-Stell-Crticlow-Fligner (DSCF pairwise comparison on the data showed significant differences between CWI and Industrial systems ($W = 4.20$, $p < 0.05$), Mobile devices ($W = 5.96$, $p < 0.01$), and smart devices ($W = 5.47$, $p < 0.05$).

6.6. Summary of Hypotheses Testing

Application security awareness ($r_s(298) = . 274$, $p < .001$) and application security self-efficacy ($r_s(298) = .3$, $p < .001$) positively correlated with information security behavior intention, while application security training ($\tau b = -.043$, $p = .466$) did not. Based on the result, Hypotheses One and Three were supported, while Hypothesis Two was unsupported. Application security self-efficacy statistically significantly differed by the deployment platform ($\chi 2$ (5) = 25.5, $p < .001$) but not by programming language ($\chi 2$ (13) = 15.6, $p > .05$). Hypothesis Five was, therefore,

supported while Hypothesis Four was unsupported. Table 6 summarizes the hypotheses testing results.

Table 6. Summary of hypotheses testing

Hypothesis	Result	Decision
H1: Application security awareness positively correlates with information security behavior intention.	Positive correlation ($r_s(298) = .274$, $p < .001$)	Supported
H2: Application security training positively correlates with information security behavior intention.	No correlation ($\tau b = -.043$, $p = .466$.).	Unsupported
H3: Application security self-efficacy positively correlates with information security behavior intention.	Positively correlated ($r_s(298) = .3$, $p < .001$).	Supported
H4: Application security self-efficacy statistically significantly differs by programming language.	No statistically significant difference ($\chi 2\ (13) = 15.6$, $p > .05$).	Unsupported
H5: Application security self-efficacy statistically significantly differs by the deployment platform.	Statistically significant difference ($\chi 2\ (5) = 25.5$, $p < .001$).	Supported

7. DISCUSSION

The surveyed software developer sample had combined poor mean SE and SeBI scores. Software developers may only be required to be proficient in some aspects of software security such as secure coding because most software security tasks require specialized skills and organizations also define specialized roles for software security tasks. However, good software developer application security self-efficacy scores are highly desirable because it leads to better developer security attitudes and more effective security implementation. Similarly, the poor mean SeBI score is undesirable because it signifies software developers' higher propensity to engage in risky security behaviors. The low SeBI score shows that software developers do not rationalize security decisions beyond coding, which aligns with Van der Linden et al. [34]. Developers need to have a good awareness of information security to value application security because application security is a subset of information security. The developers' mean SE score was greater than their mean SeBI score. This is likely because application security is more directly related to daily software developer tasks than information security.

The positive correlation between software developers' SeBI with SE and AW shows that developers' SeBI increases with their application security self-efficacy and awareness. Self-efficacy and awareness are predictors of behavioral intentions [28], [36]. The generality of self-efficacy and security awareness reflected by developers' application security self-efficacy and awareness potentially translates to increased SeBI. However, the lack of correlation between application security training and developers' SeBI was unexpected. Training is known to raise awareness. Therefore, application security training should increase application security awareness, which is positively correlated to SeBI in this study. Reasons for the lack of correlation between application security training and SeBI may be the nature and emphasis of the security training and the applicability of the knowledge to developers' daily tasks. Developer application security training should aim to increase application security awareness and self-efficacy. For these reasons, it is recommended that application security training should be relevant to developers' daily tasks and be conducted by application security professionals. Training by application security professionals increases the presence of vicarious experience and social persuasion, which are important self-efficacy sources.

The lack of statistically significant difference in application security self-efficacy scores between programming language groups was another unexpected result. However, the result can be explained by the fact that the primary security activity involving programming language is secure coding, which is the developers' primary security task. This is supported by the fact that application security self-efficacy differed by deployment platform. Software developers often are not involved in deployment platform security configurations besides those involving code. Deployment platforms also differ by security architecture and nature and prevalence of security vulnerabilities. Deployment platforms, therefore, have security requirements that software developers must learn.

8. IMPLICATIONS AND LIMITATIONS

The study's results have several practical implications. The reported application security self-efficacy and SeBI scores of software developers were poor. Software developers often prefer to delegate security responsibilities besides secure coding. Therefore, where developers are saddled with broader application security responsibilities, management must ensure that their software developers retain a high level of application security self-efficacy. Developers must also be taught to rationalize security decisions beyond coding tasks.

The study's result provides information security managers with empirical evidence of the relationship between information security behavioral intention and application security awareness, self-efficacy, and training. Applications security awareness and self-efficacy positively correlated with information security behavioral intention. Therefore, improving application security awareness and self-efficacy of software developers will improve their information security behavioral intention. Emphasizing the role application security awareness plays in general organizational information security during application security awareness training could potentially increase software developers' SeBI. The application security training recommendations from the discussion are also of value to information security managers. Application security training should be designed to be relevant and raise awareness and self-efficacy. To increase self-efficacy in application security training, the training should be hands-on and conducted by application security experts.

The study's results also inform Information security managers on the potential effect of programming languages and deployment platforms on developers' application security self-efficacy. Software developer application security self-efficacy improvement efforts do not need to place a high emphasis on the programming language used. However, the deployment platform should be emphasized in software developers' application security training and self-efficacy improvement efforts.

The study has several limitations. The study's generalization is limited to the United States, where the population of software engineers in this study resided. The study depended on a third-party audience service Pollfish to recruit the participant sample. The study is also limited to software developers. For this reason, results are not generalizable to other organizational roles, including those related to software engineering and testing. The sample size of software developers in the study is not fully representative of the United States software developers' population

9. CONCLUSION

Application security is an integral part of information security. Software controls many aspects of economic activities. As a result, the majority of security incidents result from software

vulnerabilities. Software developers are at the heart of software development. Software developers, therefore, have to be aware of application security and organizational information security. However, developers may be more aware of application security because they deal with application security issues in their programming tasks. This quantitative correlational study investigated the relationship between developers' application security-related characteristics of application security awareness, self-efficacy, and training with their information security behavior intention. Application security awareness and self-efficacy were positively correlated with information security behavior intention, while application security training did not correlate. The study also determined if software developers' application security self-efficacy differed by programming language and application deployment platform. Software developer self-efficacy statistically significantly differed by only application deployment platform. The study has practical significance and recommendations for information security managers, which were discussed. Information security managers should aim to improve software developers' application security self-efficacy and information security behavior intention. Developers should be provided with relevant hands-on application security training infused with information security awareness. Application security experts should conduct the training.

Future works could explore other software developers' application-security-related aspects, such as education, security roles performed, and years of experience. Application security training models that incorporate various elements of information security awareness could be developed to simultaneously improve software developers' application security awareness, self-efficacy, and information security behavioral intention. The effectiveness of such training models can be evaluated using pre-test and post-test score evaluations of the variables using the measurement instruments used in this study.

REFERENCES

[1] Lee, S.hyun. & Kim Mi Na, (2008) "This is my paper", ABC *Transactions on ECE*, Vol. 10, No. 5, pp120-122.

[2] Gizem, Aksahya & Ayese, Ozcan (2009) *Coomunications & Networks*, Network Books, ABC Publishers.

[3] U.S. Department of Homeland Security, "Security in the softwarelifecycle: Making software development processes—and software produced by them—more secure. DRAFT Version 1.2. ," 2006, Accessed: Jan. 28, 2023. [Online]. Available: http://www.cert.org/books/secureswe/SecuritySL.pdf.

[4] W. Umeugo, "Factors affecting the adoption of secure software practices in small and medium enterprises that build software in-house," *ijarcs*, vol. 14, no. 02, pp. 1–7, Apr. 2023, doi: 10.26483/ijarcs.v14i2.6955.

[5] Department of Homeland Security, "Software assurance," Accessed: Apr. 30, 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/infosheet_SoftwareAssurance.pdf.

[6] R. Fujdiak *et al.*, "Managing the secure software development," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Jun. 2019, pp. 1–4, doi: 10.1109/NTMS.2019.8763845.

[7] Y.-H. Tung, S.-C. Lo, J.-F. Shih, and H.-F. Lin, "An integrated security testing framework for Secure Software Development Life Cycle," in *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Oct. 2016, pp. 1–4, doi: 10.1109/APNOMS.2016.7737238.

[8] S. M. Alam, S. K. Singh, and S. A. Khan, "A Strategy Oriented Process Model for Software Security.," *International Journal of Engineering and Management Research (IJEMR)*, vol. 6, no. 6, pp. 137–142, 2016.

[9] W. C. Umeugo, "Secure software development lifecycle: A case for adoption in software SMEs," *International Journal of Advanced Research in Computer Science*, Feb. 2023.

[10] R. Rohan, D. Pal, J. Hautamäki, S. Funilkul, W. Chutimaskul, and H. Thapliyal, "A systematic literature review of cybersecurity scales assessing information security awareness.," *Heliyon*, vol. 9, no. 3, p. e14234, Mar. 2023, doi: 10.1016/j.heliyon.2023.e14234.

[11] W. M. Rodgers and L. R. Brawley, "The role of outcome expectancies in participation motivation," *Journal of Sport and Exercise Psychology*, vol. 13, no. 4, pp. 411–427, Dec. 1991, doi: 10.1123/jsep.13.4.411.

[12] S. Egelman and E. Peer, "Scaling the security wall: developing a security behavior intentions scale (sebis)," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, New York, New York, USA, Apr. 2015, pp. 2873–2882, doi: 10.1145/2702123.2702249.

[13] J. Jenkins, A. Durcikova, University of Oklahoma, USA, J. Nunamaker, and University of Arizona, USA, "Mitigating the Security Intention-Behavior Gap: The Moderating Role of Required Effort on the Intention-Behavior Relationship," *JAIS*, vol. 22, no. 1, pp. 246–272, Jan. 2021, doi: 10.17705/1jais.00660.

[14] C.-M. Chao, "Factors determining the behavioral intention to use mobile learning: an application and extension of the UTAUT model.," *Front. Psychol.*, vol. 10, p. 1652, Jul. 2019, doi: 10.3389/fpsyg.2019.01652.

[15] J. Shropshire, M. Warkentin, and S. Sharma, "Personality, attitudes, and intentions: Predicting initial adoption of information security behavior," *Computers & Security*, vol. 49, pp. 177–191, Mar. 2015, doi: 10.1016/j.cose.2015.01.002.

[16] A. Raman, R. Thannimalai, M. Rathakrishnan, and S. N. Ismail, "Investigating the influence of intrinsic motivation on behavioral intention and actual use of technology in moodle platforms," *INT. J. INSTRUCTION*, vol. 15, no. 1, pp. 1003–1024, Jan. 2022, doi: 10.29333/iji.2022.15157a.

[17] C. S. Wee, M. S. Ariff, N. Zakuan, M. N. Tajudin, K. Ismail, and N. Ishak, "Consumers perception, purchase intention and actual purchase behavior of organic food products.," *Review of Integrative Business and Economics Research*, vol. 3, no. 2, 2014.

[18] U. Konerding, "Formal models for predicting behavioral intentions in dichotomous choice situations.," *Methods of Psychological Research*, vol. 4, no. 2, pp. 1–32, 1999.

[19] M. Fishbein and I. Ajzen, "Belief, attitude, intention and behaviour: An introduction to theory and research," *Belief, attitude, intention and behaviour: An introduction to theory and research*, 1975.

[20] T. J. Madden, P. S. Ellen, and I. Ajzen, "A comparison of the theory of planned behavior and the theory of reasoned action," *Pers. Soc. Psychol. Bull.*, vol. 18, no. 1, pp. 3–9, Feb. 1992, doi: 10.1177/0146167292181001.

[21] P. Ifinedo, "Understanding information systems security policy compliance: An integration of the theory of planned behavior and the protection motivation theory," *Computers & Security*, vol. 31, no. 1, pp. 83–95, Feb. 2012, doi: 10.1016/j.cose.2011.10.007.

[22] A. Farooq, J. R. A. Ndiege, and J. Isoaho, "Factors affecting security behavior of kenyan students: an integration of protection motivation theory and theory of planned behavior," in *2019 IEEE AFRICON*, Sep. 2019, pp. 1–8, doi: 10.1109/AFRICON46755.2019.9133764.

[23] I. H. Hwang and S. H. Hu, "A study on the influence of information security compliance intention of employee: Theory of planned behavior, justice theory, and motivation theory applied.," *Journal of Digital Convergence*, vol. 16, no. 3, pp. 225–236, 2018.

[24] Y. Shih and K. Fang, "The use of a decomposed theory of planned behavior to study Internet banking in Taiwan," *Internet Research*, vol. 14, no. 3, pp. 213–223, Jul. 2004, doi: 10.1108/10662240410542643.

[25] S. H. Appelbaum and A. Hare, "Self-efficacy as a mediator of goal setting and performance," *Journal of Managerial Psych*, vol. 11, no. 3, pp. 33–47, May 1996, doi: 10.1108/02683949610113584.

[26] M. Bong and E. M. Skaalvik, :"{unav)," *Springer Science and Business Media LLC*, 2003, doi: 10.1023/a:1021302408382.

[27] R. Wood and A. Bandura, "Social cognitive theory of organizational management," *Academy of Management Review*, vol. 14, no. 3, pp. 361–384, Jul. 1989, doi: 10.5465/amr.1989.4279067.

[28] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change.," *Psychol. Rev.*, vol. 84, no. 2, pp. 191–215, 1977, doi: 10.1037/0033-295X.84.2.191.

[29] M. Kara and T. Aşti, "Effect of education on self-efficacy of Turkish patients with chronic obstructive pulmonary disease.," *Patient Educ. Couns.*, vol. 55, no. 1, pp. 114–120, Oct. 2004, doi: 10.1016/j.pec.2003.08.006.

[30] Z. Gao, P. Xiang, A. M. Lee, and L. Harrison, "Self-Efficacy and Outcome Expectancy in Beginning Weight Training Class," *Res. Q. Exerc. Sport*, vol. 79, no. 1, pp. 92–100, Mar. 2008, doi: 10.1080/02701367.2008.10599464.

[31] Bulgurcu, Cavusoglu, and Benbasat, "Information Security Policy Compliance: An Empirical Study of Rationality-Based Beliefs and Information Security Awareness," *MIS Quarterly*, vol. 34, no. 3, p. 523, 2010, doi: 10.2307/25750690.

[32] M. A. Hameed and N. A. G. Arachchilage, "The role of self-efficacy on the adoption of information systems security innovations: a meta-analysis assessment," *Pers. Ubiquitous Comput.*, vol. 25, no. 5, pp. 911–925, Oct. 2021, doi: 10.1007/s00779-021-01560-1.

[33] Md. A. Islam, M. A. Khan, T. Ramayah, and M. M. Hossain, "The Adoption of Mobile Commerce Service among Employed Mobile Phone Users in Bangladesh: Self-efficacy as a Moderator," *IBR*, vol. 4, no. 2, Mar. 2011, doi: 10.5539/ibr.v4n2p80.

[34] R. Balebako, A. Marsh, J. Lin, J. Hong, and L. Faith Cranor, "The privacy and security behaviors of smartphone app developers," presented at the Workshop on Usable Security, Reston, VA, 2014, doi: 10.14722/usec.2014.23006.

[35] R. Arizon-Peretz, I. Hadar, G. Luria, and S. Sherman, "Understanding developers' privacy and security mindsets via climate theory," *Empir. Software Eng.*, vol. 26, no. 6, p. 123, Nov. 2021, doi: 10.1007/s10664-021-09995-z.

[36] D. van der Linden *et al.*, "Schrödinger's security: Opening the box on app developers' security rationale," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, New York, NY, USA, Jun. 2020, pp. 149–160, doi: 10.1145/3377811.3380394.

[37] Jing Xie, H. R. Lipford, and Bill Chu, "Why do programmers make security errors?," in *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Sep. 2011, pp. 161–164, doi: 10.1109/VLHCC.2011.6070393.

[38] J. Zhen, K. Dong, Z. Xie, and L. Chen, "Factors influencing employees' information security awareness in the telework environment," *Electronics*, vol. 11, no. 21, p. 3458, Oct. 2022, doi: 10.3390/electronics11213458.

[39] K. Parsons, D. Calic, M. Pattinson, M. Butavicius, A. McCormac, and T. Zwaans, "The Human Aspects of Information Security Questionnaire (HAIS-Q): Two further validation studies," *Computers & Security*, vol. 66, pp. 40–51, May 2017, doi: 10.1016/j.cose.2017.01.004.

[40] K. Parsons, A. McCormac, M. Butavicius, M. Pattinson, and C. Jerram, "Determining employee awareness using the Human Aspects of Information Security Questionnaire (HAIS-Q)," *Computers & Security*, vol. 42, pp. 165–176, May 2014, doi: 10.1016/j.cose.2013.12.003.

[41] T. Moletsane and P. Tsibolane, "Mobile information security awareness among students in higher education : an exploratory study," in *2020 Conference on Information Communications Technology and Society (ICTAS)*, Mar. 2020, pp. 1–6, doi: 10.1109/ICTAS47918.2020.233978.

[42] B. Ngoqo and S. V. Flowerday, "Exploring the relationship between student mobile information security awareness and behavioural intent," *Info and Computer Security*, vol. 23, no. 4, pp. 406–420, Oct. 2015, doi: 10.1108/ICS-10-2014-0072.

[43] Cybersecurity & Infrastructure Security Agency Careers, "Secure Software Assessor ." https://www.cisa.gov/careers/work-rolessecure-software-assessor (accessed Apr. 30, 2023).

[44] D. Votipka, D. Abrokwa, and M. L. Mazurek, "Building and Validating a Scale for Secure Software Development Self-Efficacy," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Apr. 2020, pp. 1–20, doi: 10.1145/3313831.3376754.

[45] V. Hooper and C. Blunt, "Factors influencing the information security behaviour of IT employees," *Behav. Inf. Technol.*, pp. 1–13, May 2019, doi: 10.1080/0144929X.2019.1623322.

[46] G. White, T. Ekin, and L. Visinescu, "Analysis of protective behavior and security incidents for home computers," *Journal of Computer Information Systems*, vol. 57, no. 4, pp. 353–363, Oct. 2017, doi: 10.1080/08874417.2016.1232991.

[47] H.-S. Rhee, C. Kim, and Y. U. Ryu, "Self-efficacy in information security: Its influence on end users' information security practice behavior," *Computers & Security*, vol. 28, no. 8, pp. 816–826, Nov. 2009, doi: 10.1016/j.cose.2009.05.008.

[48] D.-H. (Austin) Kwak, X. Ma, and S. Kim, "When does social desirability become a problem? Detection and reduction of social desirability bias in information systems research," *Information & Management*, vol. 58, no. 7, p. 103500, Nov. 2021, doi: 10.1016/j.im.2021.103500.

[49] R. J. Fisher and J. E. Katz, "Social-desirability bias and the validity of self-reported values," *Psychol. Mark.*, vol. 17, no. 2, pp. 105–120, Feb. 2000, doi: 10.1002/(SICI)1520-6793(200002)17:2<105::AID-MAR3>3.0.CO;2-9.

[50] R. D. Hays, T. Hayashi, and A. L. Stewart, "A Five-Item Measure of Socially Desirable Response Set," *Educ. Psychol. Meas.*, vol. 49, no. 3, pp. 629–636, Sep. 1989, doi: 10.1177/001316448904900315.

[51] Y. Salem, M. Moreb, and K. S. Rabayah, "Evaluation of Information Security Awareness among Palestinian Learners," in *2021 International Conference on Information Technology (ICIT)*, Jul. 2021, pp. 21–26, doi: 10.1109/ICIT52682.2021.9491639.

[52] C. P. Dancey and J. Reidy, "Statistics without maths for psychology," *Statistics without maths for psychology*, 2007.