ESTABLISHING AND FORECASTING CYBER HACKING CONTRAVENTON

A project report submitted to Jawaharlal Nehru Technological University, Kakinada In the partial fulfilment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By
A.MRUDULA

(198A5F0002)

Under the esteem guidance of
Mr T.ESWARAIAH, M.Tech.

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

RISE KRISHNA SAI PRAKASAM GROUP OF INSTITUTION
(Affiliated to Jawaharlal Nehru Technological University, Kakinada)
Valluru-523272, Ongole, A.P.

(2019-2021)

RISE KRISHNA SAI PRAKASAM GROUP OF INSTITUTION

(Affiliated to Jawaharlal Nehru Technology University, Kakinada)
Valluru-523272, Ongole, A.P.

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

BONAFIED CERTIFICATE

This is to certify that this project work entitled "**ESTABLISHING AND FORECASTING CYBER HACKING CONTRAVENTION**"is the bonafide work carried out by **A.MRUDULA (198A5F0002)**in Partial fulfilment of the requirement for the Award of Degree in **MASTER OF COMPUTER APPLICATIONS,** and for the academic year 2019-2021. This work is done under my supervision and guidance

Internal Examiner

Mr .T.ESWARAIAH, M.Tech.

Head of the Department

Mr. T. ESWARAIAH,M.Tech,

External Examiner

ACKNOWLEDGEMENT

I express my sincere thanks to the Chairman Sri **I.C.RANGAMANNAR,** Vice Chairman Sri **S. VENKATESWARA RAO**, Secretary and Correspondent Sri **S. HANUMANTHA RAO**, and Treasurer Sri **S.BHARATH**, Rise Krishna Sai Prakasam Group of Institutions, who have provide all necessary facilities to carry out the project effectively.

Satisfaction and euphoria that accompany the successful completion of the project would be incomplete without the mention of the people who made it possible. I consider it my privilege to express my gratitude and respect to all those who guided and inspired in the successful completion of my project.

I am grateful to our Principal **Dr.A.V.BHASKAR RAO,M.Tech,Ph.D.(IIT Bombay), PDF (Univ. of Toronto, Canada) MISTE, MISET** for providing me an opportunity to do out project at RISE Krishna Sai Prakasam Group of Institutions the Valluru, Ongole and allowing availing all the faculties in the college.

I acknowledge my sincere gratitude to **Mr. T.ESWARAIAH, M.Tech,** Head of The Department of Master of Computer Applications, for his encouragement and sufficient computational facilities to successfully complete the project work.

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my guide **Mr T.ESWARAIAH, M.Tech.** Department of Master of Computer Applications for his valuable guidance throughout the course of this work. No words will be adequate to quantify their support, inspiration and cooperation. Their unflinching help, suggestion, directions and guidance have helped in progress of the project work. Their professional attitude and human qualities is a source of inspiration and model for me to follow.

I sincerely thank all the Faculty and Technical staff of Master of Computer Applications Department for their whole hearted help during the course of my work. I express ready to gratitude to my parents for encouraging me do the MCA Degree.

A.MRUDULA
198A5F0002

DECLARATION

I hereby declare that this report for the project title *"***ESTABLSHING AND FORECASTING CYBER HACKING CONTRAVENTION***"* has been developed by me under the table to supervision of Assistant Professor **Mr.T.ESWARAIAH,** M.Tech., and submitted to Department of Master of Computer Applications, Rise Krishna Sai Prakasam Group of Institutions, Valluru, Ongole for partial fulfilment of requirement of my project work. It is my own and not submitted to any other college/university or published by time**.**

Place: **A.MRUDULA**

Date: **(198A5F0002)**

ABSTRACT

Analyzing cyber incident data sets is an important method for deepening our understanding of the evolution of the threat situation. This is a relatively new research topic, and many studies remain to be done. In this paper, we report a statistical analysis of a breach incident data set corresponding to 12 years (2005–2017) of cyber hacking activities that include malware attacks. We show that, in contrast to the findings reported in the literature, both hacking breach incident *inter-arrival times* and *breach sizes* should be modeled by stochastic processes, rather Than by distributions because they exhibit auto correlations. Then, we propose particular stochastic process models to, respectively, fit the inter-arrival times and the breach sizes. We also show that these models can predict the inter-arrival times and the breach sizes. In order to get deeper insights into the evolution of hacking breach incidents, we conduct both qualitative and quantitative trend analyses on the data set. We draw a set of cyber security insights, including that the threat of cyber hacks is indeed getting worse in terms of their frequency, but not in terms of the magnitude of their damage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation		Abbreviation Description
HTML	-	Hyper Text Mark-up Language
HTTP	-	Hyper Text Transfer protocol
URL	-	Uniform Resource Locator
PHP	-	Hypertext Pre-processor
SQL	-	Structured Query Language

1 Introduction

DATA breaches are one of the most devastating cyber incidents. The Privacy Rights Clearinghouse reports 7,730 data breaches between 2005 and 2017, accounting for 9,919,228,821 breached records. The Identity Theft Resource Center and Cyber Scout reports1,093data breach incidents in 2016, which is 40% higher than the 780 data breach incidents in 2015. The United States Office of Personnel Management(OPM) reports that the personnel information of 4.2 million current and former Federal government employees and the background investigation records of current, former, and prospective federal employees and contractors (including 21.5 million Social Security Numbers) were stolen in 2015. The monetary price incurred by data breaches is also substantial. IBM reports that in year 2016, the global average cost for each lost or stolen record containing sensitive or confidential information was $158. Net Diligence.

Manuscript received November 22, 2017; revised March 16, 2018 and April 23, 2018; accepted April 28, 2018. Date of publication May 16, 2018; date of current version May 23, 2018. This work was supported in part by ARL under Grant W911NF-17-2-0127. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Conti. (Corresponding author: Shouhuai Xu.) M. Xu is with the Department of Mathematics, Illinois State University, Normal, IL 61761 USA. K. M. Schweitzer and R. M. Bateman are with the U.S. Army Research Laboratory South (Cyber), San Antonio, TX 78284 USA. S. Xu is with the Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: shxu@cs.utsa.edu).

Maillart and Sornette studied the statistical properties of the personal identity losses in the United States between year 2000 and 2008. They found that the number of breach incidents dramatically increases from 2000 to July 2006 but remains stable thereafter.

Edwards et al. analyzed a dataset containing 2,253 breach incidents that span over a decade (2005 to 2015). They found that neither the size nor the frequency of data breaches has increased over the years. Wheatley et al. analyzed a dataset that is combined from and and corresponds to organizational breach incidents between year 2000 and 2015. They found that the frequency of large breach incidents (i.e., the ones

that breach more than 50,000 records) occurring to US firms is independent of time, but the frequency of large breach incidents occurring to non-US firms exhibits an increasing trend.

The present study is motivated by several questions that have not been investigated until now, such as: Are data breaches caused by cyber-attacks increasing, decreasing, or stabilizing? A principled answer to this question will gives a clear insightinto the overallsituationof cyber threats. This question was not answered by previous studies.

Specifically, the dataset analyzed in only covered the time span from 2000 to 2008 and does not necessarily contain the breach incidents that are caused by cyber-attacks; the dataset analyzed in is more recent, but contains two kinds of incidents: negligent breaches (i.e., incidents caused by lost, discarded, stolen devices and other reasons) and malicious breaching that the other Since negligent breaches represent more human errors than cyber-attacks, we do not consider them in the present study. Because the malicious breaches studied in contain four sub-categories: hacking (including malware), insider, payment card fraud, andunknown, this study will focus on the hacking sub-category (called hacking breach dataset thereafter), while noting three sub-categories are interesting on their own and should be analyzed separately.

1.1 What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

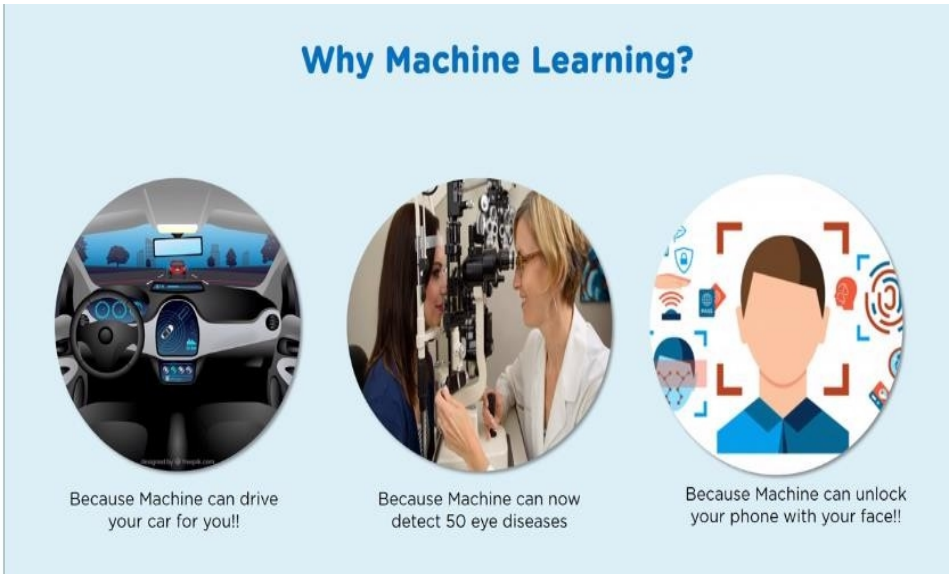


Fig 1.1: Machine Learning

1.2 Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

- Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- In contrast, unsupervised machine learning algorithms are used when the information

used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

- Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.
- Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which actions is best; this is known as the reinforcement signal.

  Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

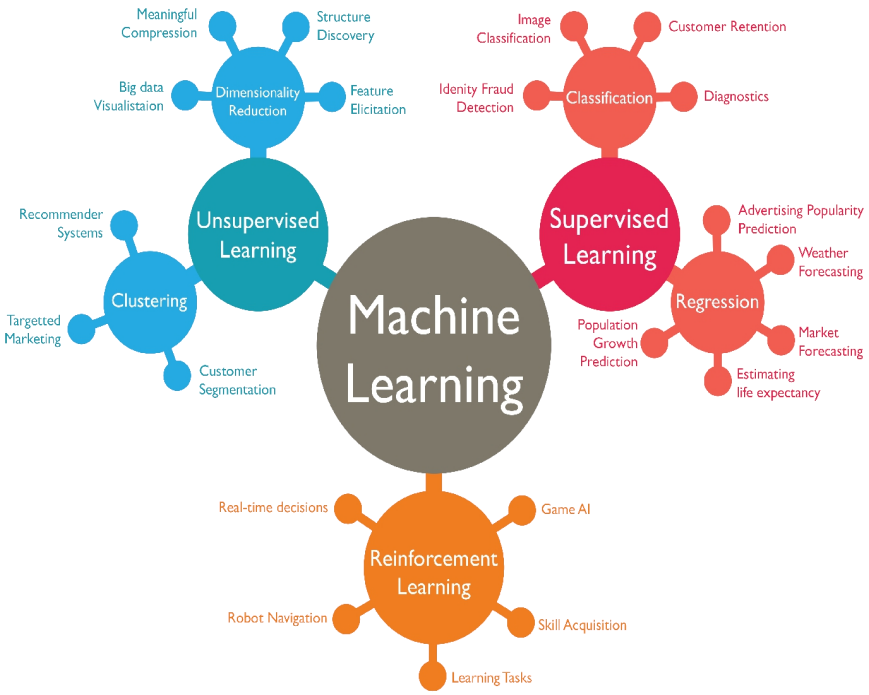


Fig 1.2: Machine learning diagram

2 Literature Survey

1.Prior Works Closely Related to the Present Study:

Maillart and Sornette analyzed a dataset of 956 personal identity loss incidents that occurred in the United States between year 2000 and 2008. They found that the personal identitylosses perincident,denotedby X, can bemodeledbya heavy tail distribution $Pr(X > n) \sim n^{-\alpha}$ where $\alpha = 0.7 \pm 0.1$. This result remains valid when dividing the dataset per type of organizations: business, education, government, and medical institution. Because the probability density function of the identity losses per incident is static, the situation of identity loss is stable from the point of view of the breach size.

Edwards et al. analyzed a different breach dataset of 2,253 breach incidents that span over a decade (2005 to 2015). These breach incidents include two categories: negligent breaches (i.e., incidents caused by lost, discarded, stolen devices, or other reasons) and malicious breaching (i.e., incidents caused by hacking, insider and other reasons). They showed that the breach size can be modeled by the log-normal or log-skewnormal distribution and the breach frequency can be modeled by the negative binomial distribution,

Wheatley et al. analyzedan organizational breach incidents dataset that is combined from and and spans over a decade (year 2000 to 2015). They used the Extreme Value Theory to study the maximum breach size, and further modeled the large breach sizes by a doubly truncated Pareto distribution. They also used linear regression to study the frequency of the data breaches, and found that the frequency of large breaching incidents is independent of time for the United States organizations, but shows an increasing trend for non-US organizations.

There are also studies on the dependenceamongcyber risks. Böhme and Kataria studied the dependence between cyber risks of two levels: within a company (internal dependence) and across companies (global dependence). Herath and Herath used the Archimedeancopula to model cyber risks caused by virus incidents, and found that there exists some dependence between these risks. Mukhopadhyay et al. used a copula-based Bayesian Belief Network to assess cyber vulnerability. Xu and Hua investigated using copulas to model dependent

cyber risks. Xu et al. used copulas to investigate the dependence encountered when modeling the effectiveness of cyber defense early-warning. Peng et al. investigated multivariate cybersecurity risks with dependence.

Compared with all these studies mentioned above, thepresent paperis uniquein that it uses a new methodologyto analyze a new perspective of breach incidents (i.e., cyber hacking breach incidents).

This perspective is important because it reflects the consequence of cyber hacking (including malware). The new methodology found for the first time, that both the incidents inter-arrival times and the breach sizes should be modeled by stochastic processes rather than distributions, and that there exists a positive dependence between them.

2) Other Prior Works Related to the Present Study:

Eling and Loperfido analyzed a dataset from the point of view of actuarial modeling and pricing. Bagchi and Udo used a variant of the Gompertz model to analyze the growth of computer and Internet-related crimes. Condon et. al used the ARIMA model to predict security incidents based on a dataset provided by the Office of Information Technology at the University of Maryland. Zhan et al. analyzed the posture of cyber threats by using a dataset collected at a network telescope.

Using datasets collected at a honeypot, Zhan et al. exploited their statistical properties including long-range dependence and extreme values to describe and predict the number of attacks against the honeypot; a predictability evaluation of a related dataset is described in. Peng et al. used a marked point process to predict extreme attack rates. Bakdash et al. extended these studies into related cybersecurity scenarios.

Liu et al. investigated how to use externally observable features of a network (e.g., mismanagement symptoms) to forecast the potential of data breach incidents to that network. Sen and Borle studied the factors that could increase or decrease the contextual risk of data breaches, by using tools that include the opportunity theory of crime, the institutional anomie theory, and the institutionaltheory.

3.1 Existing System

The present study is motivated by several questions that have not been investigated until now, such as: Are data breaches caused by cyber-attacks increasing, decreasing, or stabilizing? A principled answer to this question will give us a clear insight into the overall situation of cyber threats. This question was not answered by previous studies. Specifically, the dataset analyzed in only covered the time span from 2000 to 2008 and does not necessarily contain the breach incidents that are caused by cyber-attacks; the dataset analyzed in is more recent, but contains two kinds of incidents: negligent breaches (i.e., incidents caused by lost, discarded, stolen devices and other reasons) and malicious breaching.

Since negligent breaches represent more human errors than cyber-attacks, we do not consider them in the present study. Because the malicious breaches studied in [9] contain four sub-categories: hacking (including malware), insider, payment card fraud, and unknown, this study will focus on the hacking sub-category (called hacking breach dataset thereafter), while noting that the other three sub-categories are interesting on their own and should be analyzed separately.Recently, researchers started modeling data breach incidents. Maillart and Sornette studied the statistical properties of the personal identity losses in the United States between year 2000 and 2008. They found that the number of breach incidents dramatically increases from 2000 to July 2006 but remains stable thereafter. Edwards et al. analyzed a dataset containing 2,253 breach incidents that span over a decade (2005 to 2015). They found that neither the size nor the frequency of data breaches has increased over the years. Wheatley et al., analyzed a dataset that is combined from corresponds to organizational breach incidents between year 2000 and 2015. They found that the frequency of large breach incidents (i.e., the ones that breach more than 50,000 records) occurring to US firms is independent of time, but the frequency of large breach incidents occurring to non-US firms exhibits an increasing trend.

3.1.1 Disadvantages of Existing System

- They analyzed a dataset that is combined from corresponds to organizational breach incidents between year 2000 and 2015.
- They found that the frequency of large breach incidents occurs in independent of time.

3.2 Proposed System

In this paper, we make the following three contributions. First, we show that both the hacking breach incident interarrival times (reflecting incident frequency) and breach sizes should be modeled by stochastic processes, rather than by distributions. We find that a particular point process can adequately describe the evolution of the hacking breach incidents inter-arrival times and that a particular ARMA-GARCH model can adequately describe the evolution of the hacking breach sizes, where ARMA is acronym for "AutoRegressive and Moving Average" and GARCH is acronym for "Generalized AutoRegressive Conditional Heteroskedasticity." We show that these stochastic process models can predict the inter-arrival times and the breach sizes.

To the best of our knowledge, this is the first paper showing that stochastic processes, rather than distributions, should be used to model these cyber threat factors. Second, we discover a positive dependence between the incidents inter-arrival times and the breach sizes, and show that this dependence can be adequately described by a particular copula. We also show that when predicting inter-arrival times and breach sizes, it is necessary to consider the dependence; otherwise, the prediction results are not accurate. To the best of our knowledge, this is the first work showing the existence of this dependence and the consequence of ignoring it. Third, we conduct both qualitative and quantitative trend analyses of the cyber hacking breach incidents.

- We find that the situation is indeed getting worse in terms of the incidents inter-arrival time because hacking breach incidents become more and more frequent, but the situation is stabilizing in terms of the incident breach size, indicating that the damage of individual hacking breach incidents will not get much worse. We hope the present study will inspire more investigations, which can offer deep insights into alternate risk mitigation approaches. Such insights are useful to insurance companies, government agencies, and regulators because they need to deeply understand the nature of data breach risks.

3.2.1 Advantages of Proposed System

- We hope the present study will inspire more investigations, which can offer deep insights into alternate risk mitigation approaches.
- Such insights are useful to insurance companies, government agencies, and regulators because they need to deeply understand the nature of data breaches.

3.3 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICALFEASIBILITY
- TECHNICALFEASIBILITY
- SOCIALFEASIBILITY

3.3.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 Requirement Analysis

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

Requirement Specification

3.4.1 Functional Requirements

- Graphical User interface with the User.

3.4.2 Software Requirements

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. MySql
4. MySqlclient
5. WampServer 2.4

3.4.3 Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

3.4.4 Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 256 MB
- Space on Hard Disk: minimum 512

4.1 Architecture Diagram

Description:

Architecture is both the process and the product of planning, designing, and constructing buildings or any other structures. Architectural works, in the material form of buildings, are often perceived as cultural symbols and as works of art.

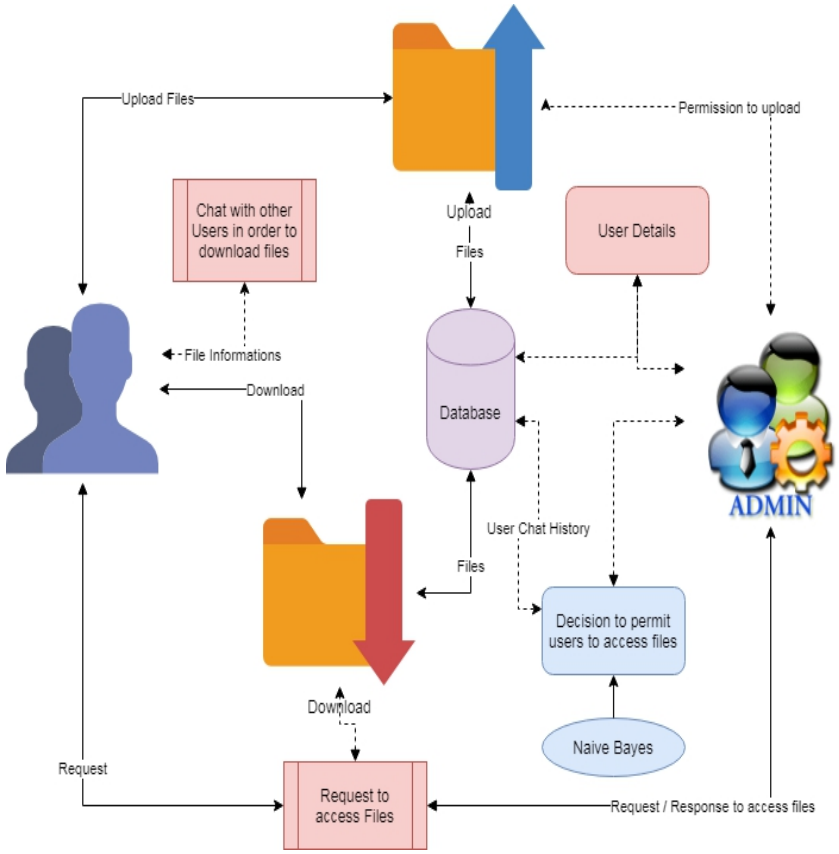


Fig 4.1: Architecture Diagram

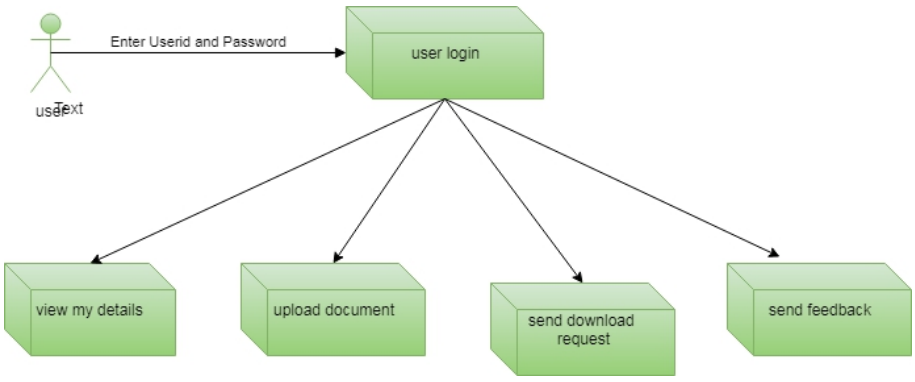
4.2 UML Diagrams

Design part is illustrated with UML Diagrams. A Diagram is the graphical presentation of asset of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). For this reason, and the UML includes nine such diagrams. The Unified Modeling Language (UML) is probably the most widely known and used notation for object- oriented analysis and design. It is the result of the merger of several early contributions to object- oriented methods. The Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts. A Modeling language is a language whose vocabulary and rules focus on the conceptual and physical representation of a system. Modeling is the designing of software applications before coding.

4.2.1 Component Diagram

- Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far.
- It does not describe the functionality of the system but it describes the components used to make those functionalities. Component-Based Development (CBD) to describe systems with Service-Oriented Architecture (SOA).

a. User



b. Admin

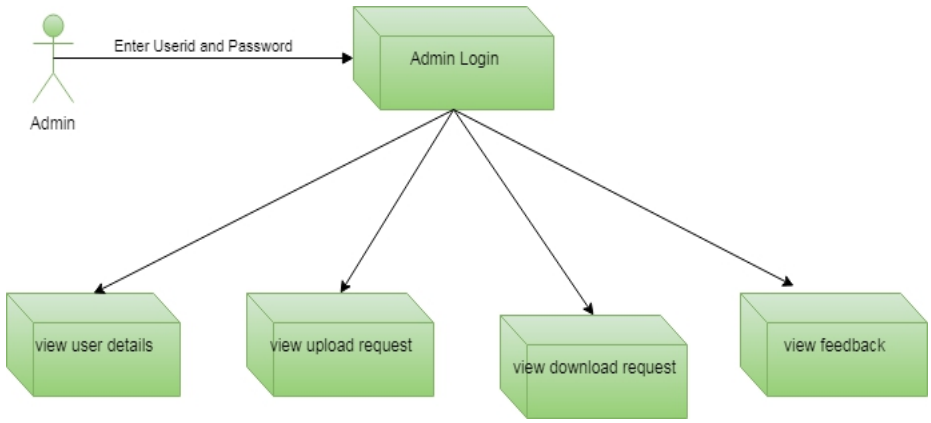
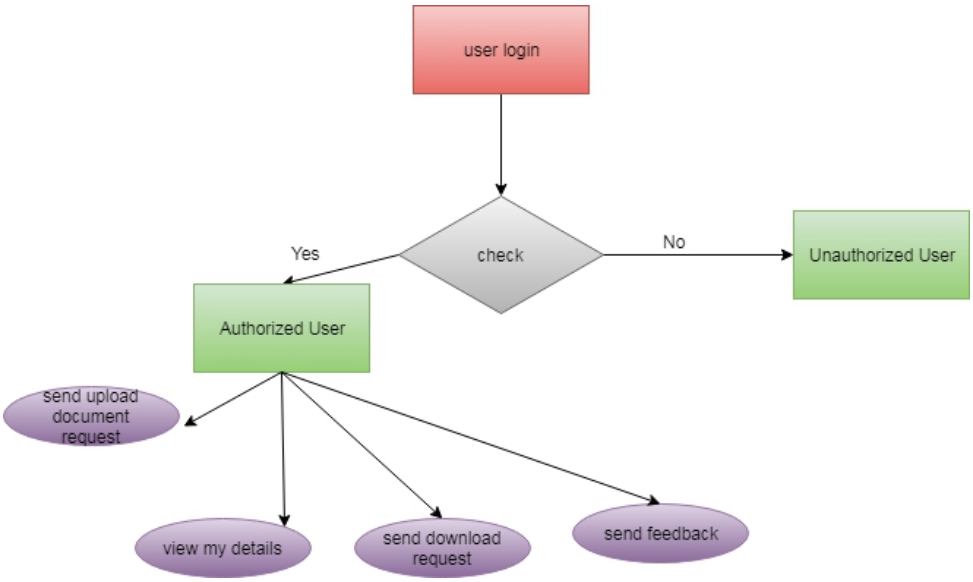


Fig 4.2.1: Component Diagram

4.2.2 Use Case Diagram

- A Use Case Diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use-cases.
- While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system

a. User



b. Admin

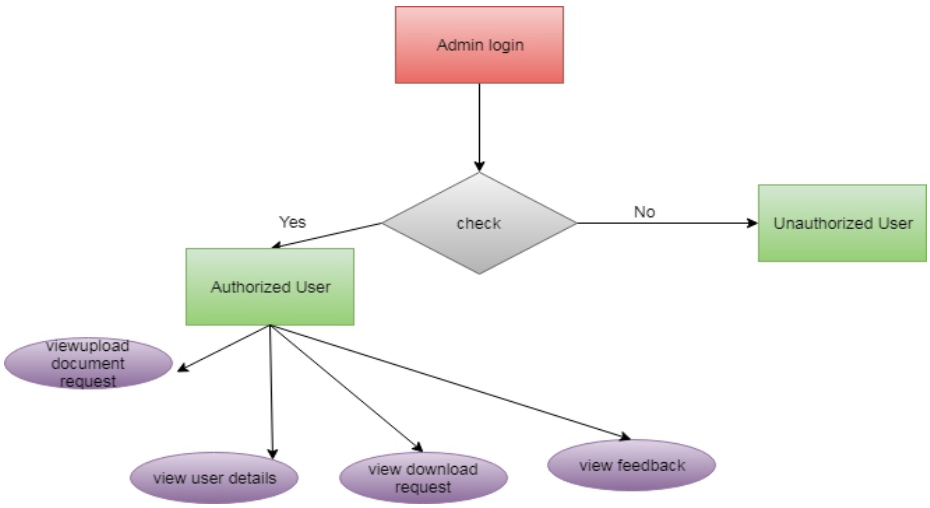


Fig 4.2.2: Use-case Diagram

4.2.3 Class Diagram

- The class diagram is used to refine the use case diagram and define a detailed design of a system. The class diagram classifies the actors defined in the use case diagram into a set of inter related classes.

- The relationship or association between the classes can be either an “is-a” or “has-a” relationship. Each class in the class diagram may be capable of providing certain functionalities.
- These functionalities provided by the class are termed “methods” of the class. Apart from this, each class may have certain “attributes” that uniquely identify the class.

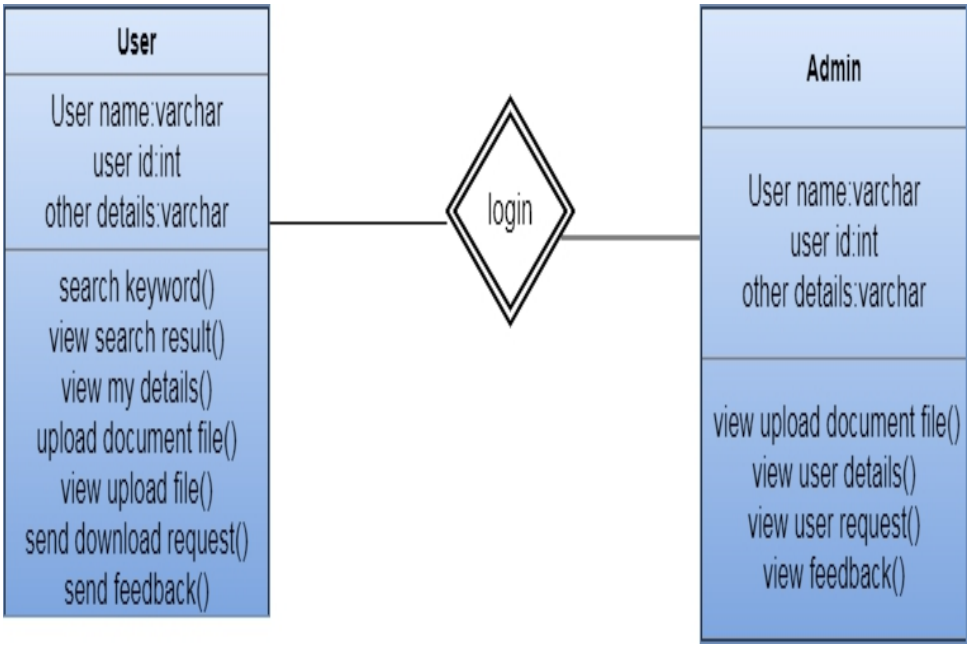
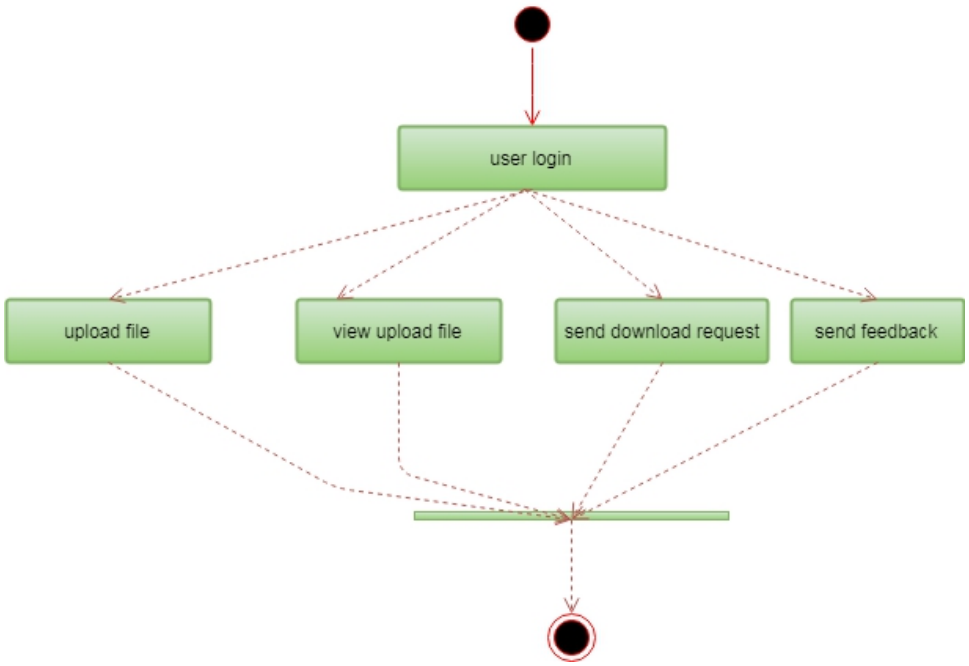


Fig 4.2.3: Class Diagram

4.2.4 Activity Diagram

- The process flows in the system are captured in the activity diagram. Similar to the state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states and guard conditions.Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
- Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.,
- Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

a. User



b. Admin

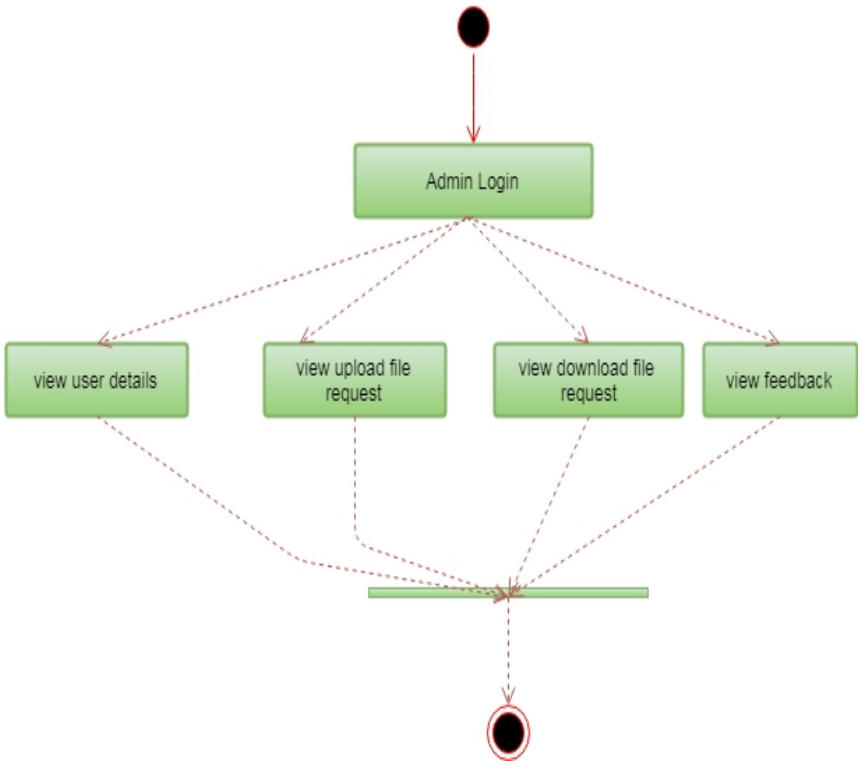
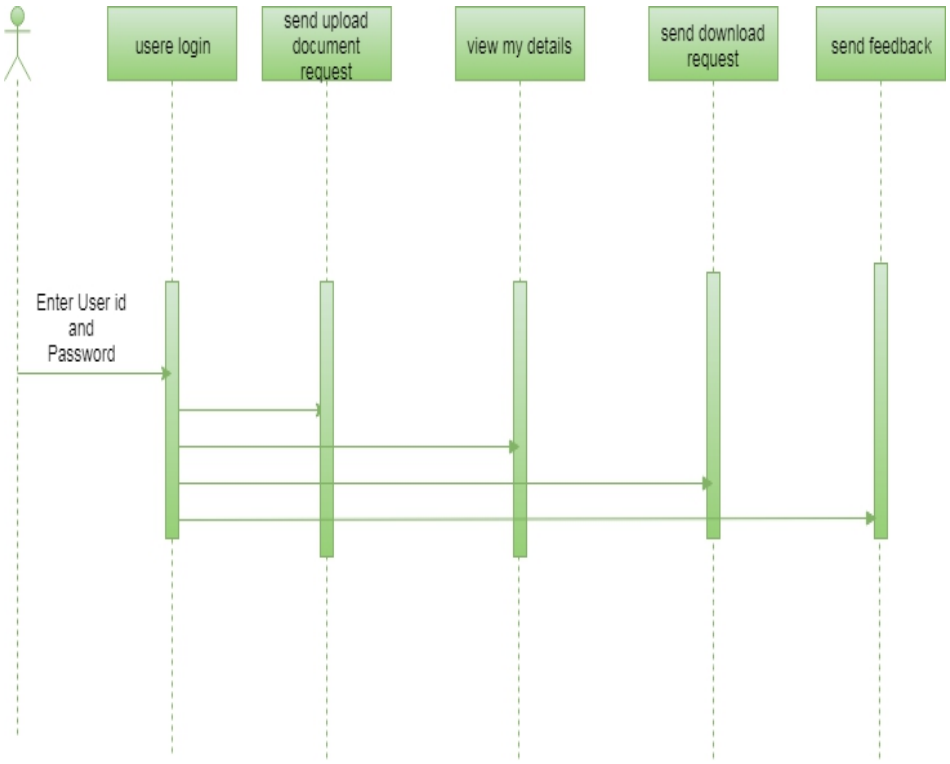


Fig 4.2.4: Activity Diagram

4.2.5 Sequence Diagram

- The sequence diagram represents the interaction between different objects in the system. The important aspect of the sequence diagram is that it is time-ordered.
- This means that the exact sequence of the interactions between the objects is represented step by step.
- Different objects in the sequence diagram interact with each other by passing the "messages"
- Sequence diagrams describe how and in what order the objects in a system function.

a. User



b. Admin

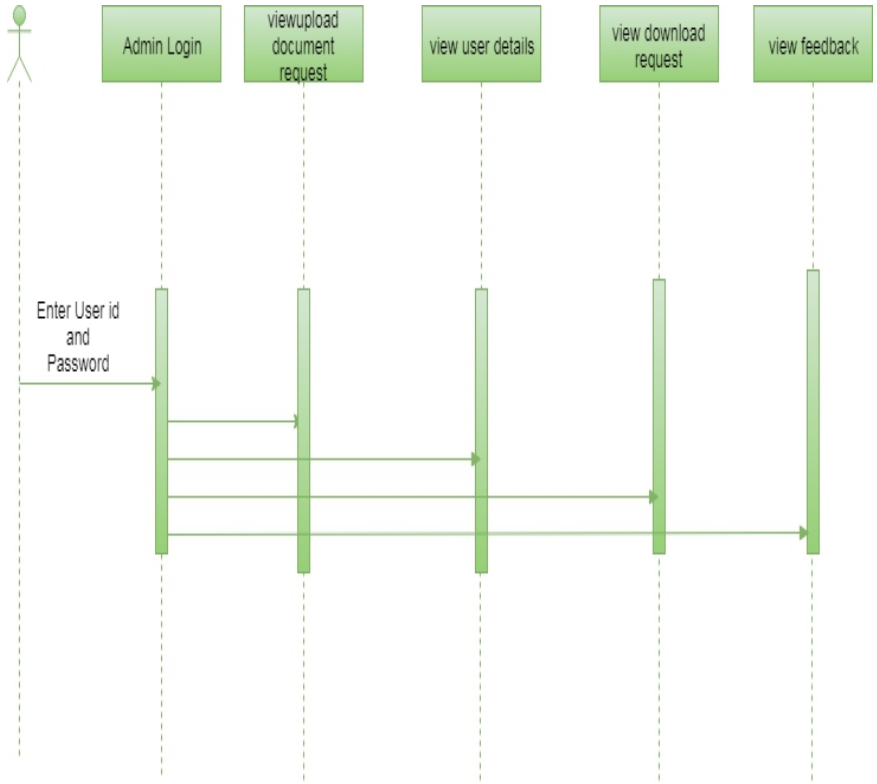
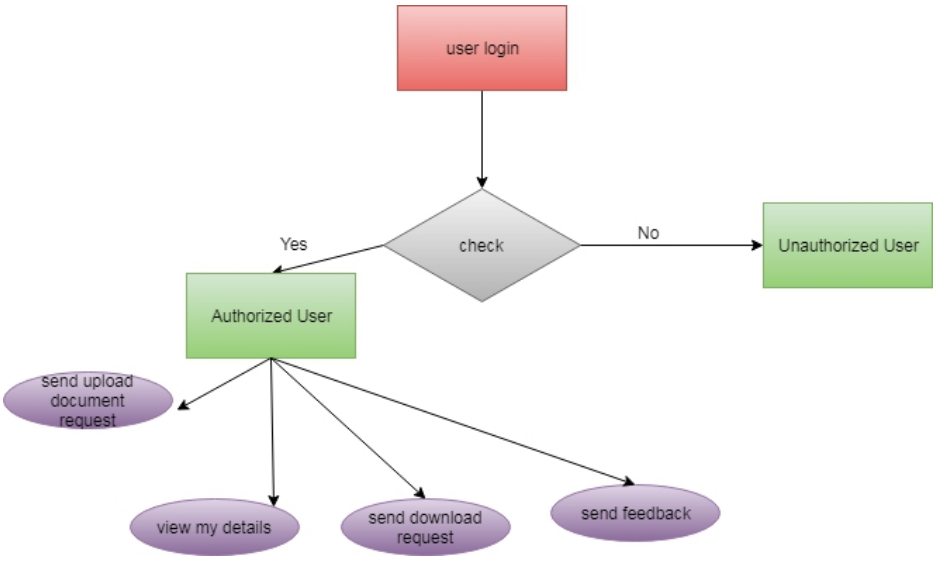


Fig 4.2.5: Sequence Diagram

4.2.6 ER Diagram

- Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design.
- An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities.
- While ER models are mostly developed for designing relational database in terms of concept visualization and in terms of physical database design, there are still other situations when ER diagrams can help.

a. User



b. Admin

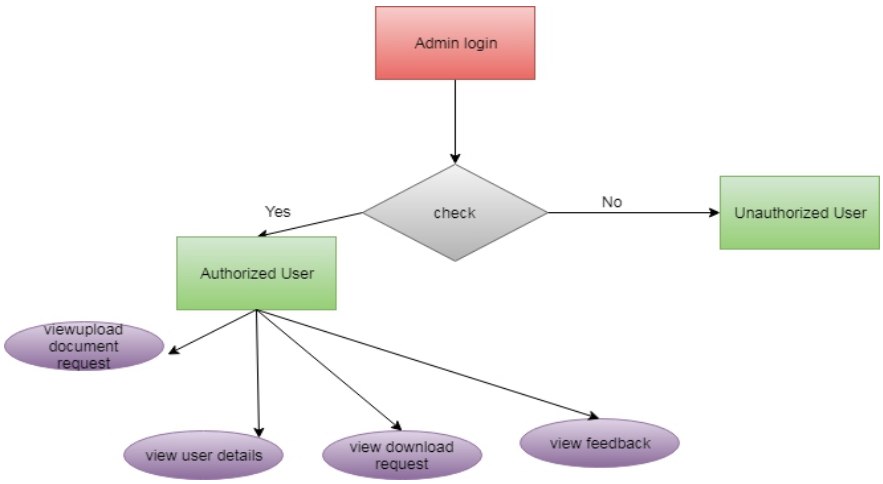


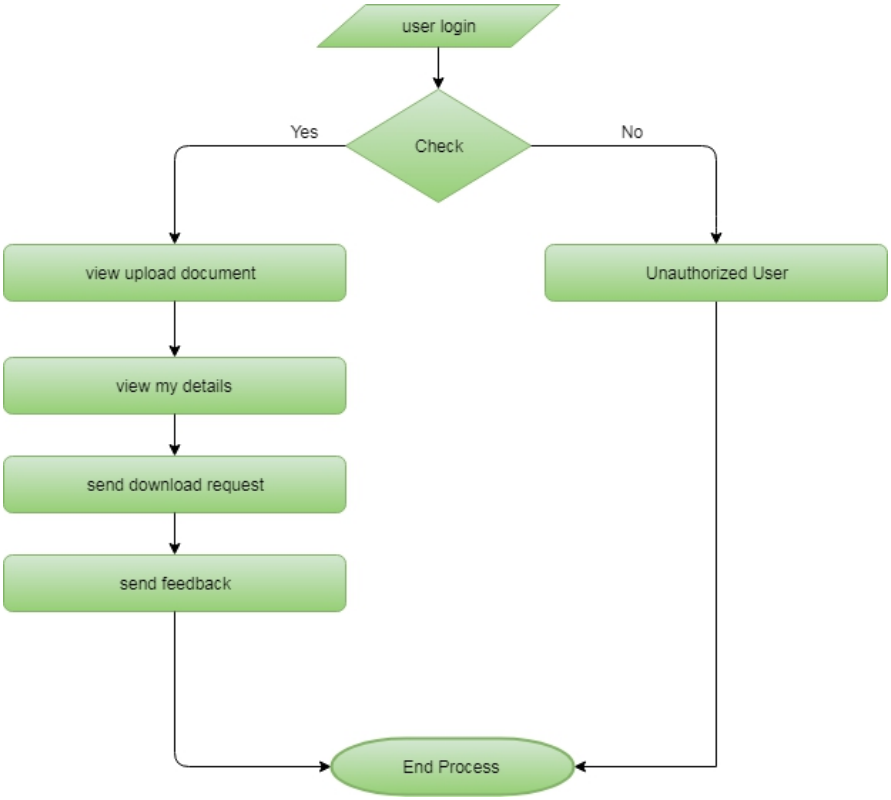
Fig 4.2.6: ER Diagram

4.2.7 Data Flow Diagram

- A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system)
- The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.
- The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram.
- Data-flow diagrams can be regarded as inverted Petri nets, because places in such networks correspond to the semantics of data memories. Analogously, the semantics of transitions from Petri nets and data flows

  and functions from data-flow diagrams should be considered equivalent.

a.User



b. Admin

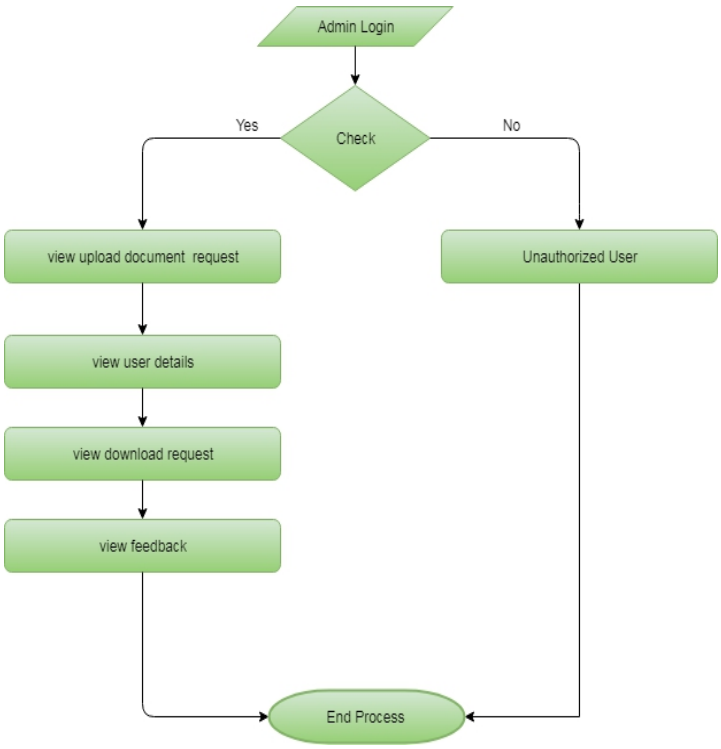


Fig 4.2.7: Data flow Diagram

5.1 Overview of System Implementation

What is python?

- hensive standard library.

- Python is a High level, structured, open-source programming language that can be used for a wide variety of programming tasks.
- Python within itself is an interpreted programming language that is automatically compiled into bytecode before execution.
- It is also a dynamically typed language that includes (but does not require one to use) object-oriented features.
- NASA has used Python for its software systems and has adopted it as the standard scripting language for its Integrated Planning System.
- Python is also extensively used by Google to implement many components of its Web Crawler and Search Engine & Yahoo! for managing its discussion groups.

History of Python

- Python was created by Guido Van Rossum.
- The design began in the late 1980s and was first released in February 1991.

Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late 70s. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

Python Version History

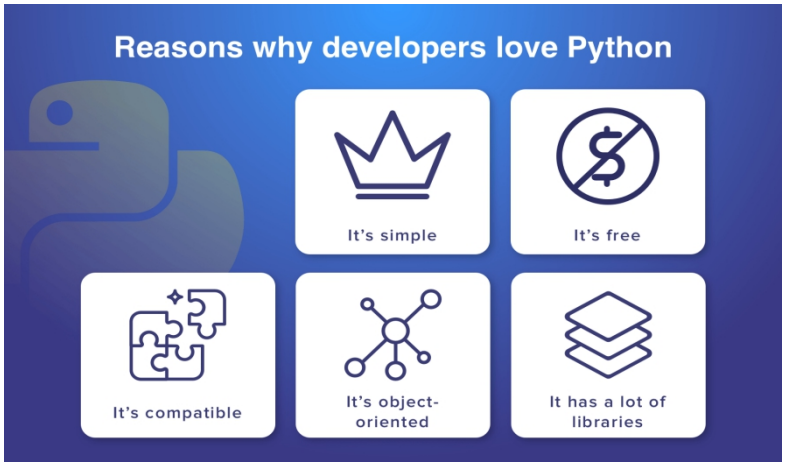
Implementation started - December 1989

Internal releases - 1990

Version No.	Date of Released
0.9	February 20, 1991
1.0	January, 1994
2.0	October 16, 2000
3.0	December 3, 2008

3.1	June 27, 2009
3.2	February 20, 2011
3.3	September 29, 2012
3.4	March 16, 2014
3.5	September 13, 2015
3.6	December 23, 2016
3.7	June 27, 2018

Features of Python Programming



What is Django?

- Django is a Web framework written in Python.
- A Web framework is a software that supports the development of dynamic Web sites, applications, and services.
- It provides a set of tools and functionalities that solves many common problems associated with Web development, such as security features, database access, sessions, template processing, URL routing, internationalization, localization, and much more.
- Using a Web framework, such as Django, enables us to develop secure and reliable Web applications very quickly in a standardized way.

The development of Django is supported by the Django Software Foundation, and it's sponsored by companies like JetBrains and Instagram.

Who's Using Django?

It's good to know who is using Django out there, so to have an idea what you can do with it. Among the biggest Web sites using Django we have: Instagram, Disqus, Mozilla, Bitbucket, Last.fm, National Geographic.

Installation

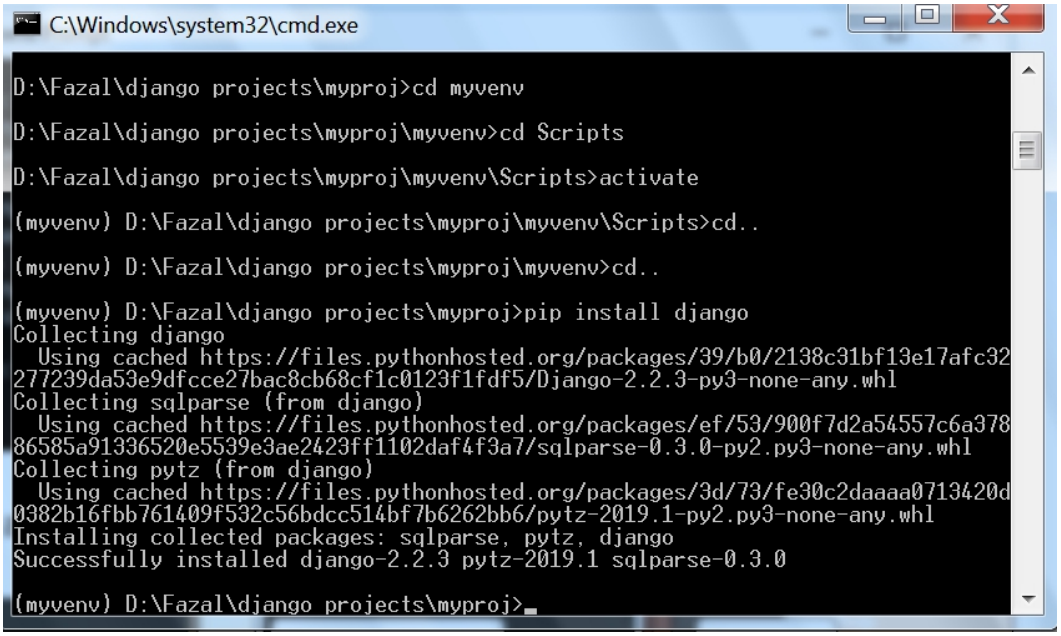
The first thing we need to do is install some programs on our machine so to be able to start playing with Django. The basic setup consists of installing

- Python
- Django

Installing Django

Now that we have the venv activated, run the following command to install Django:

```
pip install django
```



Starting a New Project

To start a new Django project, run the command below:

```
django-admin startproject myproject
```

The command-line utility django-admin is automatically installed with Django.

After we run the command above, it will generate the base folder structure for a Django project.

Our initial project structure is composed of five files:

- manage.py: a shortcut to use the django-admin command-line utility. It's used to run management commands related to our project.

  We will use it to run the development server, run tests, create migrations and much more.

- __init__.py: this empty file tells Python that this folder is a Python package.
- settings.py: this file contains all the project's configuration.
- urls.py: this file is responsible for mapping the routes and paths in our project.

  For example, if you want to show something in the URL `/about/`, you have to map it here first.

- wsgi.py: this file is a simple gateway interface used for deployment.

  You don't have to bother about it. Just let it be for now.

Django comes with a simple web server installed.

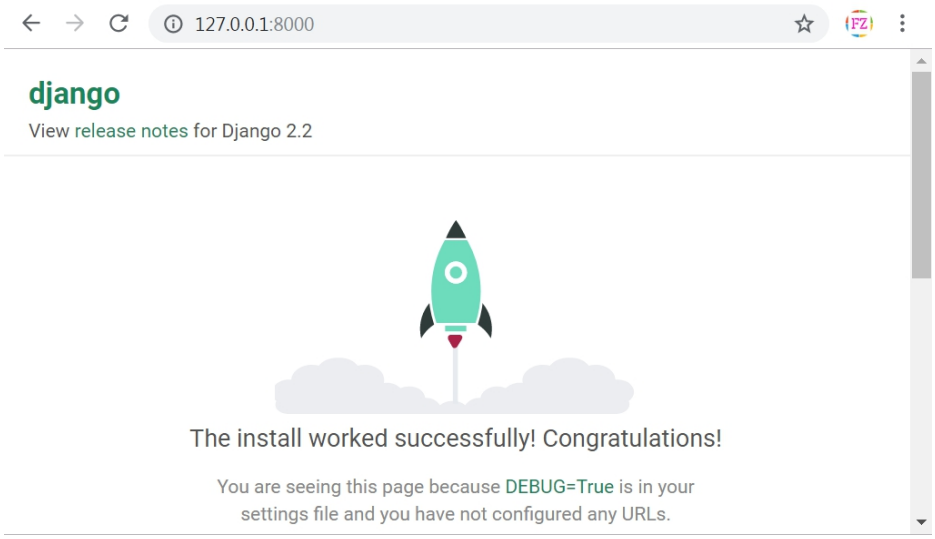
It's very convenient during the development, so we don't have to install anything else to run the project locally.

We can test it by executing the command:

```
python manage.py runserver
```

For now, you can ignore the migration errors; we will get to that later.

Now open the following URL in a Web browser: **http://127.0.0.1:8000** and you should see the following page:



Hit CTRL + BREAK to stop the development server.

Django Apps

In the Django philosophy we have two important concepts:

- app: is a Web application that does something.

  An app usually is composed of a set of models (database tables), views, templates, tests.

- project: is a collection of configurations and apps.

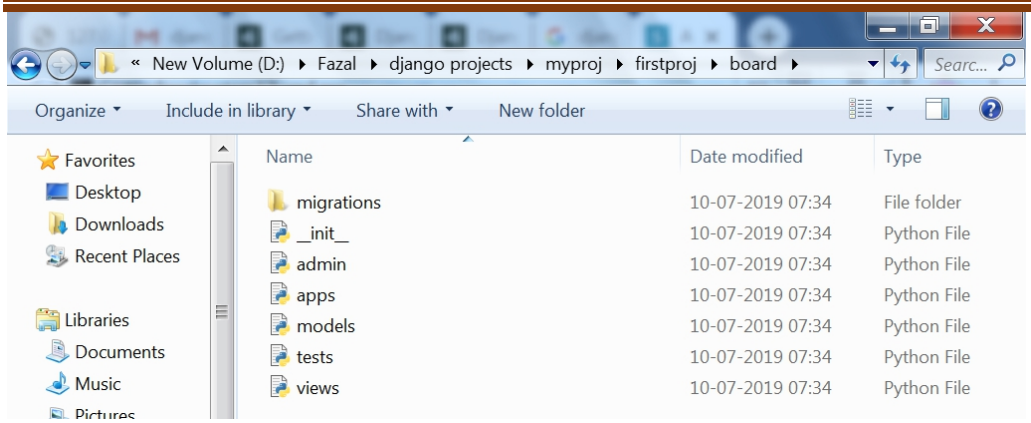
  One project can be composed of multiple apps, or a single app.

It's important to note that you can't run a Django **app** without a **project**. Simple websites like a blog can be written entirely inside a single app, which could be named **blog** or **weblog** for example.

let's create a simple Web Forum or Discussion Board. To create our first app, go to the directory where the **manage.py** file is and executes the following command:

```
django-admin startapp boards
```

Notice that we used the command **startapp** this time.



So, let's first explore what each file does:

- migrations/: here Django store some files to keep track of the changes you create in the models.py file, so to keep the database and the models.py synchronized.
- admin.py: this is a configuration file for a built-in Django app called Django Admin.
- apps.py: this is a configuration file of the app itself.
- models.py: here is where we define the entities of our Web application. The models are translated automatically by Django into database tables.
- tests.py: this file is used to write unit tests for the app.
- views.py: this is the file where we handle the request/response cycle of our Web application.

Now that we created our first app, let's configure our project to *use* it.

To do that, open the settings.py and try to find the `INSTALLED_A APPS` variable:

settings.py

```
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
```

]

As you can see, Django already come with 6 built-in apps installed. They offer common functionalities that most Web applications need, like authentication, sessions, static files management (images, javascripts, css, etc.) and so on.

Hello, World!

Let's write our first view. We will explore it in great detail in the next tutorial. But for now, let's just experiment how it looks like to create a new page with Django.

Open the views.py file inside the boards app, and add the following code:

views.py

```
fromdjango.httpimportHttpResponse
defhome(request):
returnHttpResponse('Hello, World!')
```

Views are Python functions that receive an `HttpRequest` object and returns an `HttpResponse` object. Receive a *request* as a parameter and returns a *response* as a result. That's the flow you have to keep in mind!

So, here we defined a simple view called **home** which simply returns a message saying **Hello, World!**.

Now we have to tell Django *when* to serve this view. It's done inside

the urls.py file:urls.py

```
fromdjango.conf.urlsimporturl
fromdjango.contribimportadmin
Fromboardsimportviews
urlpatterns=[
url(r'^$',views.home,name='home'),
url(r'^admin/',admin.site.urls),
]
```

If you compare the snippet above with your urls.py file, you will notice I added the following new line: `url(r'^$', views.home, name='home')` and imported the views module from our app boards using `from boards import views`.

As I mentioned before, we will explore those concepts in great detail later on.

But for now, Django works with regex to match the requested URL. For our home view, I'm using the `^$` regex, which will match an empty path, which is the homepage (this url: http://127.0.0.1:8000). If I wanted to match the URL http://127.0.0.1:8000/homepage/, my url would be:

`url(r'^homepage/$', views.home, name='home')`.

Let's see what happen:

```
python manage.py runserver
```

In a Web browser, open the http://127.0.0.1:8000 URL

5.2 Modules

1. UPLOAD DATA
2. ACCESS DETAILS
3. USER PERMISSIONS
4. DATA ANALYSIS

5.2.1 Upload Data

The data resource to database can be uploaded by both administrator and authorized user. The data can be uploaded with key in order to maintain the secrecy of the data that is not released without knowledge of user. The users are authorized based on their details that are shared to admin and admin can authorize each user. Only Authorized users are allowed to access the system and upload or request for files.

5.2.2 Access Details

The access of data from the database can be given by administrators. Uploaded data are managed by admin and admin is the only person to provide the rights to process the accessing details and approve or unapproved users based on their details.

5.2.3 User Permissions

The data from any resources are allowed to access the data with only permission from administrator. Prior to access data, users are allowed by admin to share their data and verify the details which are provided by user. If user is access the data with wrong attempts then, users are blocked accordingly. If user is requested to unblock them, based on the requests and previous activities admin is unblock users.

5.2.4 Data Analysis

Data analysis are done with the help of graph. The collected data are applied to graph in order to get the best analysis and prediction of dataset and given data policies. The dataset can be analyzed through this pictorial representation in order to better understand of the data details.

5.3 Coding

```
import re

from django.contrib import messages
from django.contrib.auth import authenticate
from django.db.models import Q, Count
from django.shortcuts import render, redirect

# Create your views here.
from Cyber_Users.forms import UserRegister_Form
from Cyber_Users.models import UserRegister_Model, UserAdd_Model


def user_login(request):
    if request.method == "POST":
        name = request.POST.get('name')
        password = request.POST.get('password')
        try:

            check = UserRegister_Model.objects.get(name=name,password=password)
            request.session['userid'] = check.id
            return redirect('user_adddata')
        except:
            pass
        user = authenticate(name=name,password=password)
        if user is not None:
            if user.is_active:


                return redirect('user_adddata')
        else:
```

```
            messages.error(request, 'username or password are not match')

            return redirect('user_login')

    return render(request, 'users/user_login.html')


def user_register(request):
    if request.method == "POST":
        forms = UserRegister_Form(request.POST)
        if forms.is_valid():
            forms.save()
            messages.success(request, 'You have been successfully registered')
            return redirect('user_login')
    else:
        forms = UserRegister_Form()

    return render(request,'users/user_register.html',{'form':forms})


def user_adddata(request):
    userid = request.session["userid"]
    obj = UserRegister_Model.objects.get(id=userid)
    attack1 = []
    attack2, attack3, attack4, attack5, attack6, attack7, attack8, attack9 = [], [], [], [], [],
[], [], [], []

    splt = ''
    Entity = ''
    Year = 0
    Records = ''
    Organizationtype = ''
    Method = ''
    txt =''
    Adddata = ''
    ans = ''
    Time = ''
    if request.method == "POST":
        Entity = request.POST.get("entity")
        Year = request.POST.get("year")
        Records = request.POST.get("records")
        Organizationtype = request.POST.get("organizationtype")
        Method = request.POST.get("method")
        txt = request.POST.get("name")
        Time = request.POST.get("time")
        splt = (re.findall(r"[\w']+", str(txt)))

    for f in splt:
        if f in ('IPid', 'FDDI', 'x25', 'rangingdistance'):
            attack1.append(f)
```

```
        elif f in ('tcpchecksum', 'mtcp', 'controlflags', 'tcpoffset', 'tcpport'):
            attack2.append(f)
        elif f in ('ICMPID', 'udptraffic', 'udpunicorn', 'datagramid', 'NTP', 'RIP', 'TFTP'):
            attack3.append(f)
        elif f in ('GETID', 'POSTID', 'openBSD', 'appid', 'sessionid', 'transid', 'physicalid'):
            attack4.append(f)
        elif f in ('SYN', 'ACK', 'synpacket', 'sycookies'):
            attack5.append(f)
        elif f in ('serverattack', 'serverid', 'blockbankwidth'):
            attack6.append(f)
        elif f in ('monlist', 'getmonlist', 'NTPserver'):
            attack7.append(f)
        elif f in ('portid', 'FTPID', 'tryion', 'fragflag'):
            attack8.append(f)
        elif f in ('malwareid', 'gethttpid', 'httpid'):
            attack9.append(f)

    if len(attack1) > len(attack2) and len(attack1) > len(attack3) and len(attack1) >
len(attack4) and len(
            attack1) > len(attack5) and len(attack1) > len(attack6) and len(attack1) >
len(attack7) and len(
        attack1) > len(attack8) and len(attack1) > len(attack9):
        ans = "Man-in-the-middle Attack"
    elif len(attack2) > len(attack1) and len(attack2) > len(attack3) and len(attack2) >
len(attack4) and len(
            attack2) > len(attack5) and len(attack2) > len(attack6) and len(attack2) >
len(attack7) and len(
        attack2) > len(attack8) and len(attack2) > len(attack9):
        ans = "Phishing and spear phishing attacks"
    elif len(attack3) > len(attack2) and len(attack3) > len(attack1) and len(attack3) >
len(attack4) and len(
            attack1) > len(attack5) and len(attack1) > len(attack6) and len(attack1) >
len(attack7) and len(
        attack1) > len(attack8) and len(attack1) > len(attack9):
        ans = "Drive-by attack"
    elif len(attack4) > len(attack2) and len(attack4) > len(attack3) and len(attack4) >
len(attack1) and len(
            attack4) > len(attack5) and len(attack4) > len(attack6) and len(attack4) >
len(attack7) and len(
        attack4) > len(attack8) and len(attack4) > len(attack9):
        ans = "Password attack"
    elif len(attack5) > len(attack2) and len(attack5) > len(attack3) and len(attack5) >
len(attack4) and len(
            attack5) > len(attack1) and len(attack5) > len(attack6) and len(attack5) >
len(attack7) and len(
        attack5) > len(attack8) and len(attack5) > len(attack9):
        ans = "SQL injection attack"
    elif len(attack6) > len(attack2) and len(attack6) > len(attack3) and len(attack6) >
len(attack4) and len(
```

```
        attack6) > len(attack5) and len(attack6) > len(attack1) and len(attack6) >
len(attack7) and len(
      attack6) > len(attack8) and len(attack6) > len(attack9):
      ans = "Cross-site scripting (XSS) attack"
       elif len(attack7) > len(attack2) and len(attack7) > len(attack3) and
len(attack7) > len(attack4) and len(
        attack7) > len(attack5) and len(attack7) > len(attack6) and len(attack7) >
len(attack1) and len(
      attack7) > len(attack8) and len(attack7) > len(attack9):
            ans = "Eavesdropping attack"
       elif len(attack8) > len(attack2) and len(attack8) > len(attack3) and
len(attack8) > len(attack4) and len(
        attack8) > len(attack5) and len(attack8) > len(attack6) and len(attack8) >
len(attack7) and len(
      attack8) > len(attack1) and len(attack8) > len(attack9):
      ans = "Birthday attack"
   elif len(attack9) > len(attack2) and len(attack9) > len(attack3) and len(attack9) >
len(attack4) and len(
        attack9) > len(attack5) and len(attack9) > len(attack6) and len(attack9) >
len(attack7) and len(
      attack9) > len(attack8) and len(attack9) > len(attack1):
      ans = "Teardrop attack"

   else:
      ans = "Unmalware"

UserAdd_Model.objects.create(uregid=obj,entity=Entity,year=Year,records=Records,
organizationtype=Organizationtype,method=Method,adddata=txt,attackresult=ans,tim
e=Time)


   return render(request,'users/user_adddata.html')

def user_page(request):
   obj = UserAdd_Model.objects.all()
   return render(request,'users/user_page.html',{'object':obj})


def malware(request):
   obj = UserAdd_Model.objects.filter(Q(attackresult='Man-in-the-middle (MitM)
attack') | Q(attackresult='Phishing and spear phishing attacks') | Q(
      attackresult='Drive-by attack') | Q(attackresult='Password attack') | Q(
      attackresult='SQL injection attack') | Q(attackresult='Cross-site scripting (XSS)
attack') | Q(attackresult='Eavesdropping attack') | Q(
      attackresult='Birthday attack') | Q(attackresult='Teardrop attack'))
   return render(request,'users/malware.html',{'object':obj})

def unmalware(request):
   obj =  UserAdd_Model.objects.filter(attackresult='Unmalware')
   return render(request,'users/unmalware.html',{'object':obj})
```

```
def breaches_analysis(request):
   chart =
UserAdd_Model.objects.values('attackresult','method').annotate(dcount=Count('attack
result'))
   return render(request,'users/breaches_analysis.html',{'objects':chart})

def chart_page(request,chart_type):
        chart =
UserAdd_Model.objects.values('year').annotate(dcount=Count('organizationtype'))
   return
render(request,'users/chart_page.html',{'chart_type':chart_type,'objects':chart})
```

5.4 Methodology

5.4.1 Support Vector Machine Algorithm

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

6.1 Overview of System Test

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 Types of Tests

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	:	identified classes of valid input must be accepted.
Invalid Input	:	identified classes of invalid input must be rejected.
Functions	:	identified functions must be exercised.
Output	:	identified classes of application outputs must be exercised.
Systems/Procedures	:	interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Box Testing Black is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

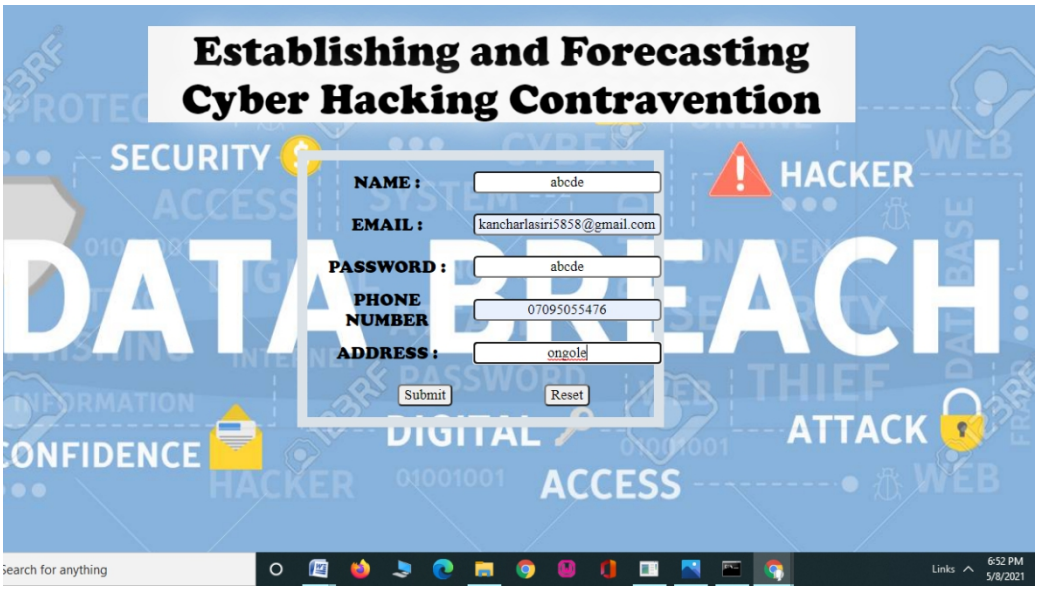
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.1 User Login Page

Description : This screenshots describes the login details of user.

7.2 User Registration page

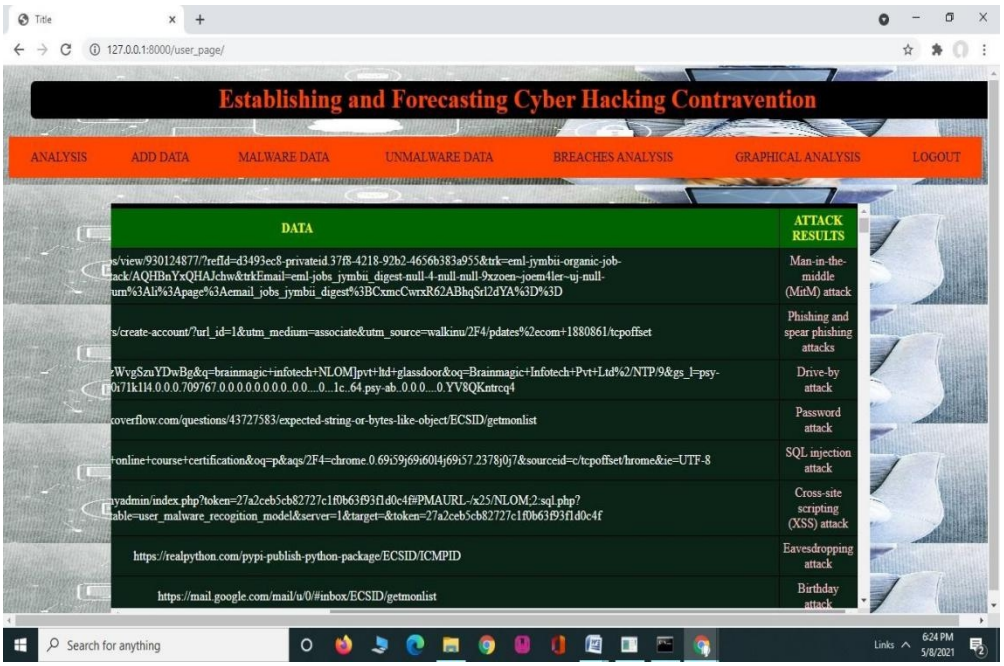


Description: This screenshot describes the filling registration details of user.

7.3 User Entering Data Page

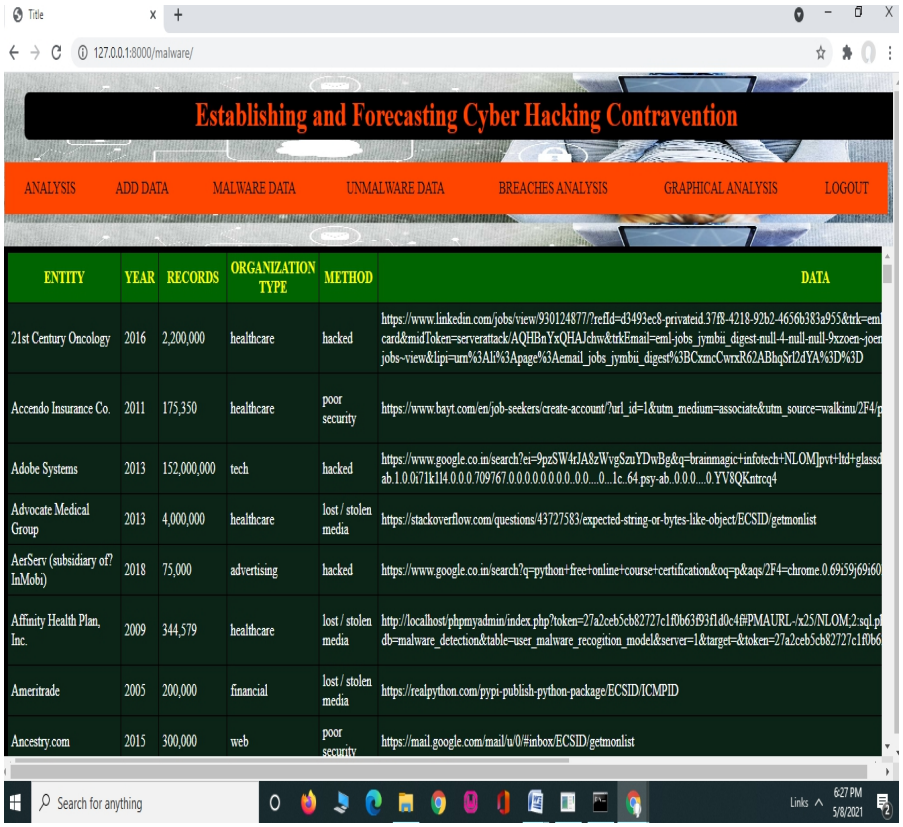
Description: This screenshot describes the entering the users data .

7.4 User checking malware data Page



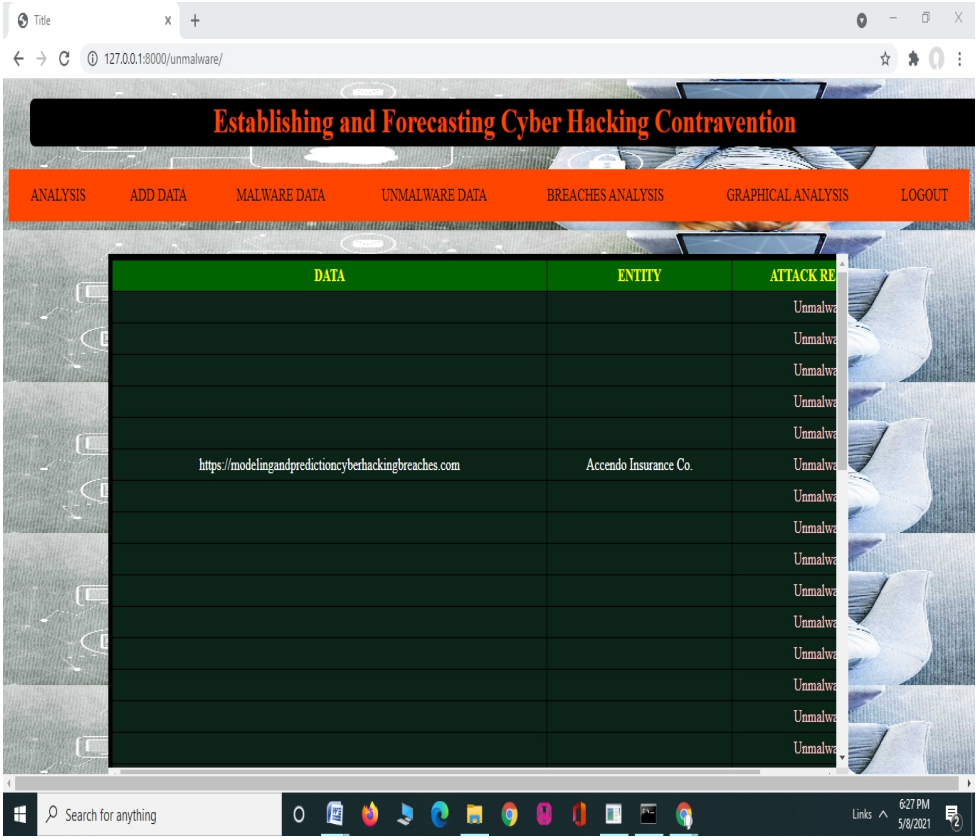
Description : This screenshot describes user checks weather the data is malware or not.

7.5Data Analysis Page



Description: This screenshot describes analyzing the malware data.

7.6 Malware Data Page

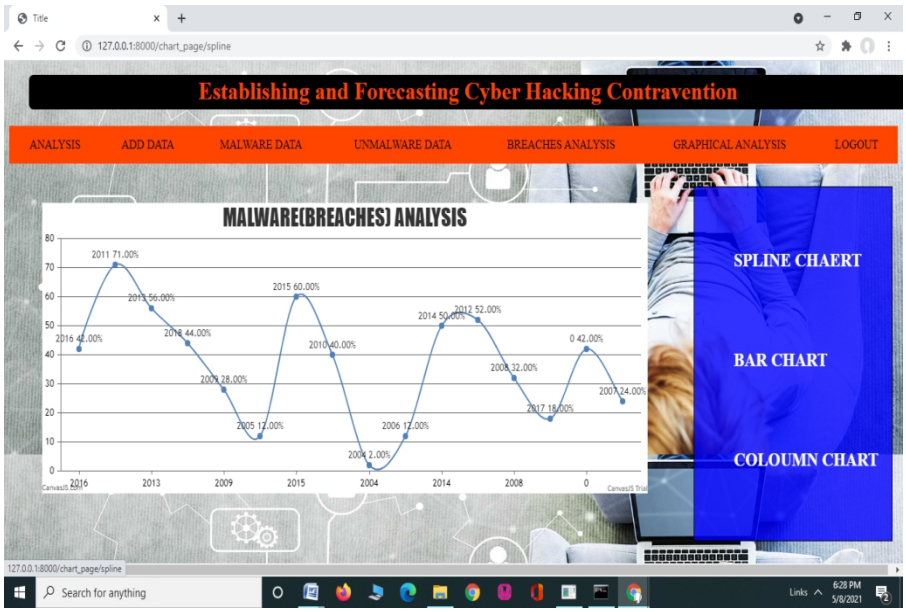


Description : This screenshot describes the entered data is a malware.

7.7 Breach Analysis Page

Description : This screenshot describes the analysis of breach incidents.

7.8 Graph Analysis

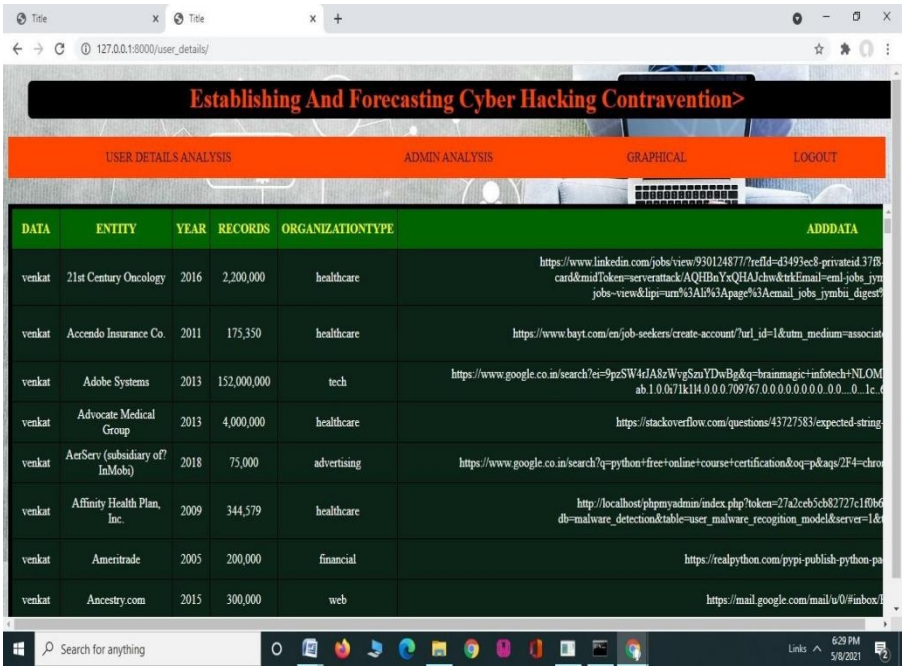


Description : This screenshot shows malware data in graphical representation.

7.9 Admin Login Page

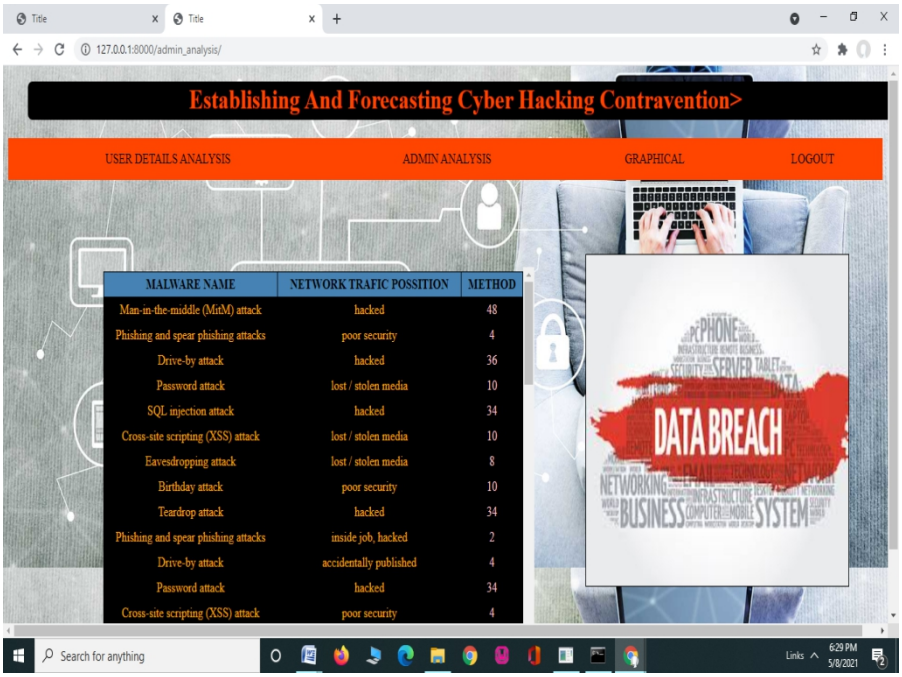
Description : This screenshot describes a login details of admin.

7.10 User details analysis



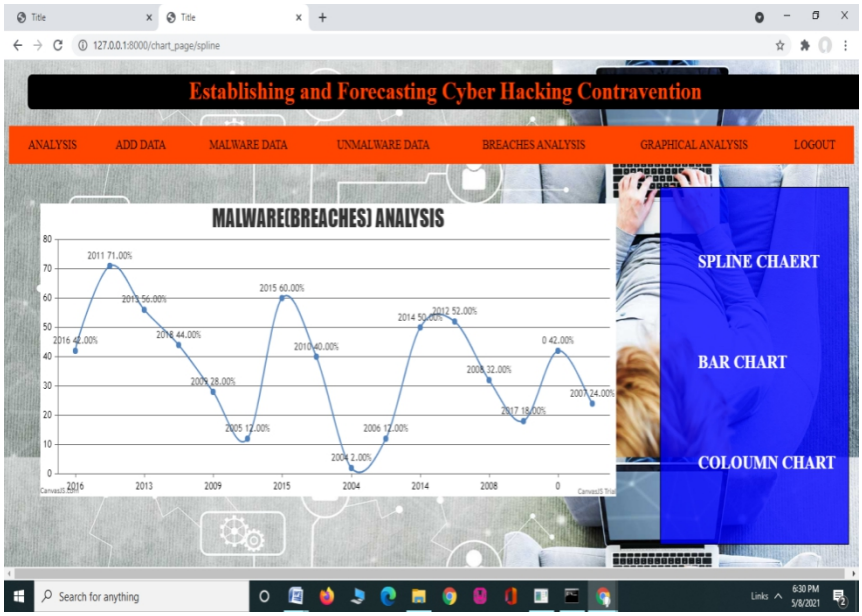
Description: This screenshot describes a analyzing a users data.

7.11 Admin Analysis Page



Description : This screenshot describes a admin analyzing a breached data.

7.12 Graphical Analysis



Description : This screenshot shows a graphical representation of malware data.

8 Conclusion

We analyzed a hacking breach dataset from the points of view of the incidents inter-arrival time and the breach size, and showed that they both should be modeled by stochastic processes rather than distributions. The statistical models developed in this paper show satisfactory fitting and prediction accuracies.

In particular, we propose using a copula-based approach to predict the joint probability that an incident with a certain magnitude of breach size will occur during a future period of time. Statistical tests show that the methodologies proposed in this paper are better than those which are presented in the literature, because the latter ignored both the temporal correlations and the dependence between the incidents inter-arrival times and the breach sizes.

We conducted qualitative and quantitative analyses to draw further insights. We drew a set of cyber security insights, including that the threat of cyber hacking breach incidents is indeed getting worse in terms of their frequency, but not the magnitude of their damage. The methodology presented in this paper can be adopted or adapted to analyze datasets of a similar nature.

9 References

[1] P. R. Clearinghouse. Privacy Rights Clearinghouse's Chronology of Data Breaches. Accessed: Nov. 2017. [Online]. Available: https://www.privacyrights.org/data-breaches.

[2] ITR Center. Data Breaches Increase 40 Percent in 2016, Finds New Report From Identity Theft Resource Center and CyberScout. Accessed: Nov. 2017. [Online]. Available: http://www.idtheftcenter.org/ 2016databreaches.html

[3] C. R. Center. Cybersecurity Incidents. Accessed: Nov. 2017. [Online]. Available: https://www.opm.gov/cybersecurity/cybersecurity-incidents.

[4] IBM Security. Accessed: Nov. 2017. [Online]. Available: https://www.ibm.com/security/data-breach/index.html.

[5] NetDiligence. The 2016 Cyber Claims Study. Accessed: Nov. 2017. [Online]. Available: https://netdiligence.com/wp-content/uploads/2016/ 10/P02_NetDiligence-2016-Cyber-Claims-Study-ONLINE.pdf.

[6] M. Eling and W. Schnell, "What do we know about cyber risk and cyber risk insurance?" J. Risk Finance, vol. 17, no. 5, pp. 474–491, 2016.

[7] T. Maillart and D. Sornette, "Heavy-tailed distribution of cyber-risks," Eur. Phys. J. B, vol. 75, no. 3, pp. 357–364, 2010.

[8] R. B. Security.Datalossdb. Accessed: Nov. 2017. [Online]. Available: https://blog.datalossdb.org.

[9] B. Edwards, S. Hofmeyr, and S. Forrest, "Hype and heavy tails: A closer look at data breaches," J. Cybersecur., vol. 2, no. 1, pp. 3–14, 2016.

[10] S. Wheatley, T. Maillart, and D. Sornette, "The extreme risk of personal data breaches and the erosion of privacy," Eur. Phys. J. B, vol. 89, no. 1, p. 7, 201

[11] P. Embrechts, C. Klüppelberg, and T. Mikosch, Modelling Extremal Events: For Insurance and Finance, vol. 33. Berlin, Germany: Springer-Verlag, 2013.

1986, pp. 97–193.

Modeling and Predicting Cyber Hacking Breaches

Maochao Xu, Kristin M. Schweitzer, Raymond M. Bateman, and Shouhuai Xu

***Abstract*— Analyzing cyber incident data sets is an important method for deepening our understanding of the evolution of the threat situation. This is a relatively new research topic, and many studies remain to be done. In this paper, we report a statistical analysis of a breach incident data set corresponding to 12 years (2005–2017) of cyber hacking activities that include malware attacks. We show that, in contrast to the findings reported in the literature, both hacking breach incident *inter-arrival times* and *breach sizes* should be modeled by stochastic processes, rather than by distributions because they exhibit autocorrelations. Then, we propose particular stochastic process models to, respectively, fit the inter-arrival times and the breach sizes. We also show that these models can predict the inter-arrival times and the breach sizes. In order to get deeper insights into the evolution of hacking breach incidents, we conduct both qualitative and quantitative trend analyses on the data set. We draw a set of cybersecurity insights, including that the threat of cyber hacks is indeed getting worse in terms of their frequency, but not in terms of the magnitude of their damage.**

***Index Terms*— Hacking breach, data breach, cyber threats, cyber risk analysis, breach prediction, trend analysis, time series, cybersecurity data analytics.**

I. Introduction

DATA breaches are one of the most devastating cyber incidents. The Privacy Rights Clearinghouse [1] reports 7,730 data breaches between 2005 and 2017, accounting for 9,919,228,821 breached records. The Identity Theft Resource Center and Cyber Scout [2] reports 1,093 data breach incidents in 2016, which is 40% higher than the 780 data breach incidents in 2015. The United States Office of Personnel Management (OPM) [3] reports that the personnel information of 4.2 million current and former Federal government employees and the background investigation records of current, former, and prospective federal employees and contractors (including 21.5 million Social Security Numbers) were stolen in 2015. The monetary price incurred by data breaches is also substantial. IBM [4] reports that in year 2016, the global average cost for each lost or stolen record containing sensitive or confidential information was \$158. NetDiligence [5] reports that in year 2016, the median number of breached records was 1,339, the median per-record cost was \$39.82, the average breach cost was \$665,000, and the median breach cost was \$60,000.

While technological solutions can harden cyber systems against attacks, data breaches continue to be a big problem. This motivates us to characterize the evolution of data breach incidents. This not only will deep our understanding of data breaches, but also shed light on other approaches for mitigating the damage, such as insurance. Many believe that insurance will be useful, but the development of accurate cyber risk metrics to guide the assignment of insurance rates is beyond the reach of the current understanding of data breaches (e.g., the lack of modeling approaches) [6].

Recently, researchers started modeling data breach incidents. Maillart and Sornette [7] studied the statistical properties of the personal identity losses in the United States between year 2000 and 2008 [8]. They found that the number of breach incidents dramatically increases from 2000 to July 2006 but remains stable thereafter. Edwards *et al.* [9] analyzed a dataset containing 2,253 breach incidents that span over a decade (2005 to 2015) [1]. They found that neither the size nor the frequency of data breaches has increased over the years. Wheatley *et al.* [10] analyzed a dataset that is combined from [8] and [1] and corresponds to organizational breach incidents between year 2000 and 2015. They found that the frequency of large breach incidents (i.e., the ones that breach more than 50,000 records) occurring to US firms is independent of time, but the frequency of large breach incidents occurring to non-US firms exhibits an increasing trend.

The present study is motivated by several questions that have not been investigated until now, such as: *Are data breaches caused by cyber attacks increasing, decreasing, or stabilizing?* A principled answer to this question will give us a clear insight into the overall situation of cyber threats. This question was not answered by previous studies. Specifically, the dataset analyzed in [7] only covered the time span from 2000 to 2008 and does not necessarily contain the breach incidents that are caused by cyber attacks; the dataset analyzed in [9] is more recent, but contains two kinds of incidents: *negligent breaches* (i.e., incidents caused by lost, discarded, stolen devices and other reasons) and *malicious breaching*. Since negligent breaches represent more human errors than cyber attacks, we do not consider them in the present study. Because the malicious breaches studied in [9] contain four sub-categories: *hacking (including malware)*, *insider*, *payment card fraud*, and *unknown*, this study will focus on the *hacking* sub-category (called *hacking breach* dataset thereafter), while noting that the other three sub-categories are interesting on their own and should be analyzed separately.

Manuscript received November 22, 2017; revised March 16, 2018 and April 23, 2018; accepted April 28, 2018. Date of publication May 16, 2018; date of current version May 23, 2018. This work was supported in part by ARL under Grant W911NF-17-2-0127. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mauro Conti. *(Corresponding author: Shouhuai Xu.)*

M. Xu is with the Department of Mathematics, Illinois State University, Normal, IL 61761 USA.

K. M. Schweitzer and R. M. Bateman are with the U.S. Army Research Laboratory South (Cyber), San Antonio, TX 78284 USA.

S. Xu is with the Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: shxu@cs.utsa.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2834227

A. Our Contributions

In this paper, we make the following three contributions. First, we show that both the hacking breach incident *inter-arrival times* (reflecting incident *frequency*) and *breach sizes* should be modeled by stochastic processes, rather than by distributions. We find that a particular point process can adequately describe the evolution of the hacking breach incidents inter-arrival times and that a particular ARMA-GARCH model can adequately describe the evolution of the hacking breach sizes, where ARMA is acronym for "AutoRegressive and Moving Average" and GARCH is acronym for "Generalized AutoRegressive Conditional Heteroskedasticity." We show that these stochastic process models can predict the inter-arrival times and the breach sizes. To the best of our knowledge, this is the first paper showing that stochastic processes, rather than distributions, should be used to model these cyber threat factors.

Second, we discover a positive dependence between the incidents inter-arrival times and the breach sizes, and show that this dependence can be adequately described by a particular copula. We also show that when predicting inter-arrival times and breach sizes, it is necessary to consider the dependence; otherwise, the prediction results are not accurate. To the best of our knowledge, this is the first work showing the existence of this dependence and the consequence of ignoring it.

Third, we conduct both qualitative and quantitative trend analyses of the cyber hacking breach incidents. We find that the situation is indeed getting worse in terms of the incidents inter-arrival time because hacking breach incidents become more and more frequent, but the situation is stabilizing in terms of the incident breach size, indicating that the damage of individual hacking breach incidents will not get much worse.

We hope the present study will inspire more investigations, which can offer deep insights into alternate risk mitigation approaches. Such insights are useful to insurance companies, government agencies, and regulators because they need to deeply understand the nature of data breach risks.

B. Related Work

1) Prior Works Closely Related to the Present Study: Maillart and Sornette [7] analyzed a dataset [8] of 956 personal identity loss incidents that occurred in the United States between year 2000 and 2008. They found that the personal identity losses per incident, denoted by X, can be modeled by a heavy tail distribution $\Pr(X > n) \sim n^{-\alpha}$ where $\alpha = 0.7 \pm 0.1$. This result remains valid when dividing the dataset per type of organizations: business, education, government, and medical institution. Because the probability density function of the identity losses per incident is static, the situation of identity loss is stable from the point of view of the breach size.

Edwards *et al.* [9] analyzed a different breach dataset [1] of 2,253 breach incidents that span over a decade (2005 to 2015). These breach incidents include two categories: *negligent breaches* (i.e., incidents caused by lost, discarded, stolen devices, or other reasons) and *malicious breaching* (i.e., incidents caused by hacking, insider and other reasons). They showed that the breach size can be modeled by the log-normal or log-skewnormal distribution and the breach frequency can be modeled by the negative binomial distribution, implying that neither the breach size nor the breach frequency has increased over the years.

Wheatley *et al.* [10] analyzed an organizational breach incidents dataset that is combined from [8] and [1] and spans over a decade (year 2000 to 2015). They used the Extreme Value Theory [11] to study the maximum breach size, and further modeled the large breach sizes by a doubly truncated Pareto distribution. They also used linear regression to study the frequency of the data breaches, and found that the frequency of large breaching incidents is independent of time for the United States organizations, but shows an increasing trend for non-US organizations.

There are also studies on the dependence among cyber risks. Böhme and Kataria [12] studied the dependence between cyber risks of two levels: within a company (internal dependence) and across companies (global dependence). Herath and Herath [13] used the Archimedean copula to model cyber risks caused by virus incidents, and found that there exists some dependence between these risks. Mukhopadhyay *et al.* [14] used a copula-based Bayesian Belief Network to assess cyber vulnerability. Xu and Hua [15] investigated using copulas to model dependent cyber risks. Xu *et al.* [16] used copulas to investigate the dependence encountered when modeling the effectiveness of cyber defense early-warning. Peng *et al.* [17] investigated multivariate cybersecurity risks with dependence.

Compared with all these studies mentioned above, the present paper is unique in that it uses a new methodology to analyze a new perspective of breach incidents (i.e., cyber hacking breach incidents). This perspective is important because it reflects the consequence of cyber hacking (including malware). The new methodology found for the first time, that both the incidents inter-arrival times and the breach sizes should be modeled by stochastic processes rather than distributions, and that there exists a positive dependence between them.

2) Other Prior Works Related to the Present Study: Eling and Loperfido [18] analyzed a dataset [1] from the point of view of actuarial modeling and pricing. Bagchi and Udo [19] used a variant of the Gompertz model to analyze the growth of computer and Internet-related crimes. Condon *et. al* [20] used the ARIMA model to predict security incidents based on a dataset provided by the Office of Information Technology at the University of Maryland. Zhan *et al.* [21] analyzed the posture of cyber threats by using a dataset collected at a network telescope. Using datasets collected at a honeypot, Zhan *et al.* [22], [23] exploited their statistical properties including long-range dependence and extreme values to describe and predict the number of attacks against the honeypot; a predictability evaluation of a related dataset is described in [24]. Peng *et al.* [25] used a marked point process to predict extreme attack rates. Bakdash *et al.* [26] extended these studies into related cybersecurity scenarios. Liu *et al.* [27] investigated how to use externally observable features of a network (e.g., mismanagement symptoms) to forecast the potential of data breach incidents to that network. Sen and Borle [28] studied the factors that could increase or decrease the contextual risk of data breaches, by using tools that include the opportunity theory of crime, the institutional anomie theory, and the institutional theory.

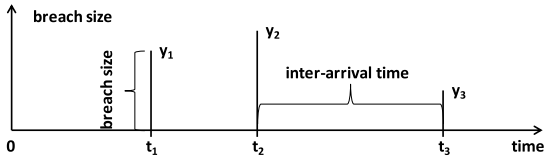


Fig. 1. Illustrative description of cyber hacking breach incidents.

C. Paper Outline

The rest of the paper is organized as follows. In Section II we describe the dataset and research questions. In Section III we present a basic analysis of the dataset. In Section IV we develop a novel point process model for analyzing the dataset. In Section V, we discuss the prediction performance of the proposed model. In Section VI we present qualitative and quantitative trend analyses. In Section VII we conclude our paper with future research directions. We defer formal description of the main statistical notions to the Appendix, and discuss their intuitive meanings when they are mentioned for the first time.

II. Research Questions and Dataset Description

A. Research Questions

Figure 1 gives an illustrative description of cyber hacking breach incidents. There are three incidents that occur respectively at times t_1, t_2, and t_3, each exposing a different number of data records. The incidents are *irregularly* spaced because $t_2 - t_1 \neq t_3 - t_2$. Two concepts of interest are: the *inter-arrival times* between two consecutive incidents, which lead to a time series $\{d_1 = t_1, d_2 = t_2 - t_1, d_3 = t_3 - t_2, \ldots\}$; and the *breach sizes* (i.e., the number of data records that are compromised because of an incident), which lead to a time series $\{y_1, y_2, y_3, \ldots\}$.

Given a dataset of cyber hacking breach incidents, we want to use it to answer the following questions.

1) Should we use a *distribution* or *stochastic process* to describe the breach incidents inter-arrival times, and which distribution or process? This question is important because answering it will directly deepen our understanding of the dynamic cyber hacking breach situation from a *temporal* perspective. (Sections III and IV)
2) Should we use a distribution or stochastic process to describe the breach sizes, and which distribution or process? This question is important because answering it will directly deepen our understanding of the dynamic cyber hacking breach situation from a *magnitude* perspective. (Sections III and IV)
3) Are the breach sizes and the incidents inter-arrival times independent of each other? If not, how should we characterize the dependence between them? This question is important because answering it will directly deepen our understanding of the dynamic cyber hacking breach situation from a joint *temporal* and *magnitude* perspective. (Section IV)
4) Can we predict when the next hacking incident will occur, and what the breach size would be? This question is important because answering it shows our capability to *predict* the situation and possibly conduct *proactive defense* at a small time scale (e.g., days or weeks ahead of time). For example, when the probability that a big breach incident will occur during the next week is high, the defender may dynamically adjust the defense posture (e.g., enforcing more restricted policies during the next week). This is similar to what weather forecasting can do in the physical world. (Section V)
5) What are the trends that are exhibited by hacking breach incidents? This question is important because we can draw higher-level insights into whether the situation is getting better or worse over a large time scale (e.g., 10 years), and to what extent. (Section VI)

B. Dataset

The hacking breach dataset we analyze in this paper was obtained from the Privacy Rights Clearinghouse (PRC) [1], which is the largest and most extensive dataset that is also publicly available. Since we focus on hacking breaches, we disregard the negligent breaches and the other sub-categories of malicious breaches (i.e., insider, payment card fraud, and unknown). From the remaining raw hacking breaches data, we further disregard the incomplete records with unknown/unreported/missing hacking breach sizes because breach size is one of the objects for our study.

The resulting dataset contains 600 hacking breach incidents in the United States between January 1st, 2005 and April 7th, 2017. The hacking breach victims span over 7 industries: businesses-financial and insurance services (BSF); businesses-retail/merchant including online retail (BSR); businesses-other (BSO); educational institutions (EDU); government and military (GOV); healthcare, medical providers and medical insurance services (MED); and nonprofit organizations (NGO).

The dataset is represented by a sequence, denoted by $\{(t_i, y_{t_i})\}_{0 \leq i \leq 600}$, where t_i represents the day on which there is an incident of breach size y_{t_i} (i.e., the number of private data records that are breached by the incident), and t_0 is the day on which observation starts (i.e., t_0 does not correspond to the occurrence of any incident). The inter-arrival times are $d_i = t_i - t_{i-1}$, where $i = 1, 2, \ldots, 600$. Among the t_i's, most days have one single incident report, 52 days with 2 incidents on each day, 7 days with 3 incidents on each day, and one day (02/26/2016) with 7 incidents.

We caution that the dataset does not necessarily contain all of the hacking breach incidents, because there may be unreported ones. Moreover, the dates corresponding to the incidents are the days on which the incidents are reported, rather than the dates on which the incidents took place. Nevertheless, this dataset (or data source [1]) represents the best dataset that can be obtained in the public domain [9], [29]. Therefore, analysis of it will shed light on the severeness of the data breach risk, and the analysis methodologies can be adopted or adapted to analyze more accurate datasets of this kind when they become available in the future.

C. Preprocessing

Because we observed, as mentioned above, some days have multiple hacking breach incidents, one may suggest to treat such multiple incidents as a single "combined" incident (i.e., adding their number of breached records together).

TABLE I
SUMMARY OF NOTATIONS (r.v. STANDS FOR RANDOM VARIABLE)

t	time, which is used when describing a general model
$C(\cdot)$	copula function, which is used to model the dependence
$\{(t_i, y_{t_i})\}_i$	the ith incident occurring at time t_i with breach size y_{t_i}
d_i	breach incidents inter-arrival time $d_i = t_i - t_{i-1}$
$\mathrm{VaR}_\alpha(t)$	the Value-at-Risk at level $0 < \alpha < 1$ for r.v. X_t: $\mathrm{VaR}_\alpha(t) = \inf\{l : P(X_t \leq l) \geq \alpha\}$

However, this method is not sound because the multiple incidents may happen to different victims that have different cyber systems. Given that the time resolution of the dataset is a day, multiple incidents that are reported on the same data may be reported at different points in time of the same day (e.g., 8pm vs. 10pm). As such, we propose generating small random time intervals to separate the incidents corresponding to the same day. Specifically, we randomly order the incidents corresponding to the same day, and then insert a small and random time interval in between two consecutive incidents (for the first interval, the starting point is midnight), while assuring that these incidents correspond to the same day (e.g., the two incidents on a two-incident day may be assigned at 8am and 1pm).

D. Remark

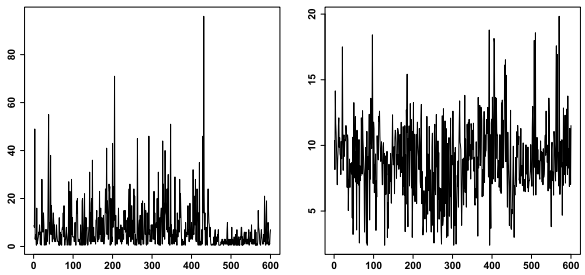
In this paper, we use a number of statistical techniques, a thorough review of which would be lengthy. In order to comply with the space requirement, here we only briefly review these techniques at a high level, and refer the readers to specific references for each technique when it is used. We use the *autoregressive conditional mean* point process [30], [31], which was introduced for describing the evolution of conditional means, to model the evolution of the inter-arrival time. We use the ARMA-GARCH time series model [32], [33] to model the evolution of the breach size, where the ARMA part models the evolution of the mean of the breach sizes and the GARCH part models the high volatility of the breach sizes. We use copulas [34], [35] to model the nonlinear dependence between the inter-arrival times and the breach sizes.

Table I summarizes the main notations used in the paper.

III. BASIC ANALYSIS

Figure 2 plots the two time series that are actually investigated in the present paper. Figure 2(a) plots the time series of incidents inter-arrival time (unit: day). We observe that most inter-arrival times are small (say, less than 20 days), and that the recent inter-arrival times are even smaller, which hints that the frequency of hacking breaches intensifies. That is, Figure 2(a) hints the existence of clusters of small inter-arrival times (i.e., multiple incidents occur during a short period of time). One possible explanation for the cluster phenomenon is the following: multiple successful hacks are detected and reported within a very short period of time because the attackers used the same attacks or exploited the same vulnerabilities, which are detected at roughly the same time. Figure 2(a) also shows that the breach incidents are irregularly spaced (i.e., exhibiting both large and small inter-arrival times).

Figure 2(b) plots the log-transformed breach sizes (unit: record) because the breach sizes exhibit large variability and



(a) (b)

Fig. 2. Time series plots of inter-arrival times and log-transformed breach sizes of the aggregated incidents (x-axis is the sequence of incidents). Fig 2(a) shows that more recent breach incidents have smaller inter-arrival times. Fig 2(b) shows that there is a huge volatility in the breach size. (a) Incidents inter-arrival times (y-axis with a unit 'day'). (b) Log-transformed breach sizes (y-axis with a unit 'record').

TABLE II
STATISTICS OF BREACH INCIDENTS INTER-ARRIVAL TIME (UNIT: DAY), WHERE 'SD' STANDS FOR STANDARD DEVIATION

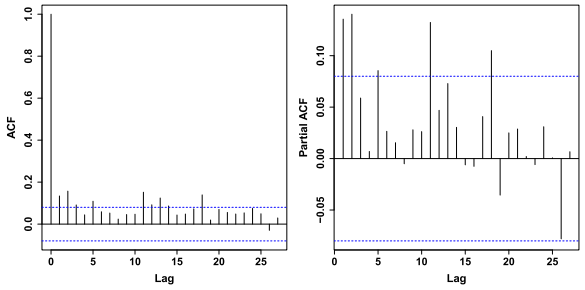
Victim category	Min	Median	Mean	SD	Max	Total
BSF	.0255	28.00	61.9200	75.1958	378	69
BSO	.0254	28.00	52.8900	71.8083	451	84
BSR	1.00	37.50	66.30	84.9736	447	60
EDU	.0227	14.00	26.260	36.4019	256	165
GOV	2.00	44.50	89.58	109.4744	455	50
MED	.0258	3.00	27.50	72.2903	497	163
NGO	67.0	203.00	376.3	382.7538	1178	9
Aggregate	.0026	2.00	4.00	7.4710	96	600

skewness, which make it difficult to model the breach sizes. We observe a large volatility in the breach size and the volatility clustering phenomenon of large (small) changes followed by large (small) changes. We also observe that some breach sizes are especially large (meaning severe hacking breach incidents). We will pay particular attention for modeling these extreme breach incidents.

A. Basic Analysis of Breach Incidents Inter-Arrival Times

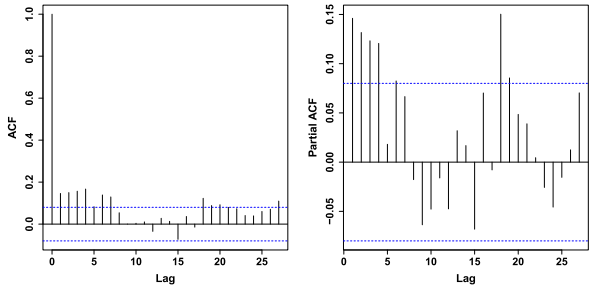
Table II describes the basic statistics of the inter-arrival times for individual victim categories as well as the aggregation of them (which corresponds to Figure 2). We observe that the standard deviation of the inter-arrival times in each category is also much larger than the mean, which hints that the processes describing the hacking breach incidents are not Poisson. We also observe that the *aggregation* of the inter-arrival times of all categories leads to much smaller inter-arrival times. For example, the maximum inter-arrival time of NGO breach incidents is 1178 days, while the maximum inter-arrival time of the aggregation is 96 days.

In order to formally answer the question whether the incidents inter-arrival times should be modeled by a distribution or a stochastic process, we look into the sample AutoCorrelation Function (ACF) and Partial AutoCorrelation Function (PACF) of the inter-arrival times. Intuitively, ACF measures the correlation between the observations at earlier times and the observations at later times *without* disregarding



(a) (b)

Fig. 3. The sample ACF and PACF of incidents inter-arrival times. (a) ACF of inter-arrival times. (b) PACF of inter-arrival times.



(a) (b)

Fig. 4. The sample ACF and PACF of log-transformed breach sizes. (a) ACF of transformed breach sizes. (b) PACF of transformed breach sizes.

TABLE III
STATISTICS OF HACKING BREACH SIZES, WHERE 'SD' STANDS FOR STANDARD DEVIATION

Size	Min	Median	Mean	SD	Max	Total
BSF	11	2000	2228000	10436141	76000000	69
BSO	11	6470	9677000	49488457	412000000	84
BSR	12	1464	2666000	14678814	100000000	60
EDU	20	10870	41940	95481.03	800000	165
GOV	24	14000	119400	293147.3	1700000	50
MED	180	4668	34140	96820.77	697600	163
NGO	444	15000	28190	34754.27	110000	9
Total	11	6324	1909000	19588938	412000000	600

the observations in between them, and PACF measures the correlation between the observations at earlier times and the observations at later times *while* disregarding the observations in between them. The formal definitions of ACF and PACF are given in Appendix A. ACF and PACF are widely used to detect temporal correlations in time series [36], [37].

Figure 3 plots the sample ACF and PACF, respectively. We observe correlations in both plots because there are correlation values that exceed the dashed blue lines (i.e., the threshold values which are derived based on the asymptotic statistical theory [36], [38]). This means that there are significant correlations between the inter-arrival times and that the inter-arrival times do not follow the exponential distribution. Moreover, we should use a stochastic process to describe the inter-arrival times [39]. In summary, we have:

Insight 1: The hacking breach incidents inter-arrival times exhibit some clusters of small inter-arrival times (i.e., multiple incidents occur within a short period of time) and the incidents are irregularly spaced. Moreover, there are correlations between the inter-arrival times, meaning that the inter-arrival times should be modeled by an appropriate stochastic process rather than by a distribution.

B. Basic Analysis of Hacking Breach Sizes

Table III summarizes the basic statistics of the hacking breach sizes. We observe that three Business categories have much larger mean breach sizes than others. We further observe that there exists a large standard deviation for the breach size in each of the victim categories, and that the standard deviation is always much larger than the corresponding mean. Figure 2(b) plots the log-transformed breach sizes because, as we can observe from Table III, the breach sizes exhibit large volatility and skewness (which is indicated by the substantial difference between the median and the mean values), which make them hard to model without making transformations.

In order to answer the question whether the breach sizes should be modeled by a distribution or stochastic process, we plot the temporal correlations between the breach sizes. Figures 4(a) and 4(b) plot the sample ACF and PACF for the log-transformed breach sizes, respectively. We observe correlations between the breach sizes, meaning that we should use a stochastic process, rather than a distribution, to model the breach sizes [33], [36]. This is in contrast to the insight offered by previous studies [7], [18], which suggests to use a skewed distribution to model the breach sizes. We attribute the drawing of this insight to the fact that these studies [7], [18] did *not* look into this due perspective of temporal correlations. An important factor for determining whether to use a distribution or a stochastic process to describe something, depends on whether or not there is temporal autocorrelation between the individual samples. This is because zero temporal autocorrelation means that the samples are independent of each other; otherwise, non-zero temporal autocorrelation means that they are not independent of each other and should not be modeled by a distribution.

Insight 2: The hacking breach sizes exhibit a large volatility, a large skewness, and a volatility clustering phenomenon, namely large (small) changes followed by large (small) changes. Moreover, there are correlations between the breach sizes, implying that they should be modeled by an appropriate stochastic process than a distribution.

IV. MODELING THE HACKING BREACH DATASET

In this section, we develop a novel statistical model to fit the breach dataset, or more specifically the *in-sample* of 320 incidents. The fitted model will be used for prediction, which will be evaluated by the *out-of-sample* of 280 incidents (Section V).

A. Modeling the Inter-Arrival Times

Insight 1 suggests that we model the hacking breach incidents inter-arrival times with an *autoregressive conditional mean* (ACD) model, which was originally introduced to model the evolution of the inter-arrival time, or *duration*, between

stock transactions [30] and later extended to model duration processes (see, e.g., [31], [40]).[1]

Recall that the dataset is represented by a sequence $\{(t_i, y_{t_i})\}_{0\le i\le n}$, where $n = 600$, t_i for $i \ge 1$ is the day on which there is an incident of breach size y_{t_i}. The inter-arrival times are $d_i = t_i - t_{i-1}$, where $i = 1, 2, \ldots, n$. The basic idea of the *conditional mean* model is to standardize the inter-arrival time $d_i = t_i - t_{i-1}$ by leveraging the historic information, where $i = 1, 2, \ldots, n$. Specifically, we define

$$d_i = \Psi_i \epsilon_i, \tag{IV.1}$$

where the Ψ_i's are functions of the historical inter-arrival times

$$\Psi_i = \mathrm{E}(d_i|\mathfrak{F}_{i-1})$$

with $\mathfrak{F}_{i-1}$ representing the historical information up to time t_{i-1}, and the ϵ_i's are independent and identically distributed (i.i.d.) innovations with $\mathrm{E}(\epsilon_i) = 1$.

1) Model Selection: For model selection, we focus on the following ACD models because (i) these models are relatively simple and can be efficiently estimated in practice; and (ii) these models are flexible enough to accommodate the evolution of the inter-arrival times based on our preliminary analysis.

- The standard ACD model (ACD) [30]:

$$\Psi_i = \omega + \sum_{j=1}^{p} a_j d_{i-j} + \sum_{j=1}^{q} b_j \Psi_{i-j},$$

  where subscript i indicates the ith breach incident, $\omega, a_j, b_j \ge 0$, and p and q are positive integers indicating the orders of the autoregressive terms.
- The type-I log-ACD model (LACD_1) [41]:

$$\log(\Psi_i) = \omega + \sum_{j=1}^{p} a_j \log(\epsilon_{i-j}) + \sum_{j=1}^{q} b_j \log(\Psi_{i-j}).$$

- The type-II log-ACD model (LACD_2) [41]:

$$\log(\Psi_i) = \omega + \sum_{j=1}^{p} a_j \log(d_{i-j}) + \sum_{j=1}^{q} b_j \log(\Psi_{i-j}).$$

In what follows, we further restrict our investigation to the case of $p = q = 1$ because a higher order does not necessarily improve the prediction accuracy [42]. The distribution of the standardized innovations of the ϵ_i's is assumed to be a generalized gamma distribution. This assumption will be validated below. We make this assumption because it is flexible and because it was recommended in the literature for modeling irregularly spaced data [40], [42].

Recall that the density function of the generalized gamma distribution is

$$f(x|\lambda, \gamma, k) = \frac{\gamma\, x^{k\gamma-1}}{\lambda^{k\gamma}\,\Gamma(k)} \exp\left\{-\left(\frac{x}{\lambda}\right)^{\gamma}\right\}, \tag{IV.2}$$

where $\lambda > 0$ is the scale parameter, and $\gamma, k > 0$ are the shape parameters. The generalized gamma distribution includes many well-known distributions as special cases, such as the exponential distribution, the Weibull distribution, the half-normal distribution, and the gamma distribution. In order to assure $E(\epsilon_i) = 1$, in our estimation we set

$$\lambda = \frac{\Gamma(k)}{\Gamma(k + 1/\gamma)}.$$

We use the maximum likelihood estimation (MLE) method [31] to fit the model parameters. Table IV describes the fitting results. We observe that according to the model selection methods Akaikes Information Criterion (AIC) and Bayes Information Criterion (BIC) [36], which (as reviewed in Appendix B) intuitively measure how well the proposed model fit the observations (i.e., the smaller these values, the better the fitting), LACD_1 should be selected. We also observe that the coefficient $b_1 = -0.767$ (0.0971) of LACD_1 is statistically significant, where 0.0971 is the estimated standard deviation. This means that the historic inter-arrival times do have a significant effect on the current inter-arrival time. We further observe that $k\gamma < 1$ and $\gamma > 1$, implying that the conditional hazard function of inter-arrival times is U-shaped.

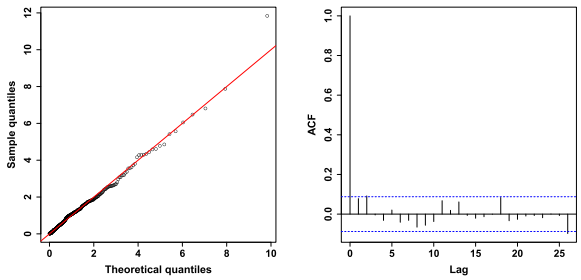
In order to formally evaluate the fitting accuracy of LACD_1, we plot the fitting residuals in Figure 5. Figure 5(a) is the qq-plot of the residuals, and shows that all points except one are around the 45-degree line, meaning that the fitting is accurate.

In order to examine whether or not the proposed LACD_1 model is sufficient to capture the dependence between the inter-arrival times, we plot the sample ACF of the residuals in Figure 5(b), which shows that the correlations at all lags are very small. In particular, the right-hand half of Table V presents the p-values of the formal McLeod-Li and Ljung-Box statistical tests [31], [36], which (as reviewed in Appendix C) intuitively measure whether or not there are correlations that are left in the residuals. We observe that these

TABLE IV

MODEL FITTING RESULTS OF THE ACD AND LOG-ACD MODELS TO THE INTER-ARRIVAL TIMES OF HACKING BREACH INCIDENTS. THE NUMBERS IN THE PARENTHESES ARE THE ESTIMATED STANDARD DEVIATIONS

Model	ω	a_1	b_1	k	γ	AIC	BIC
ACD	3.0559 (3.367)	0.0682 (0.065)	0.5705 (0.427)	0.5802 (0.121)	1.2061 (0.170)	1997.81802 -	2016.65963 -
LACD1	**3.825** (0.2254)	**0.058** (0.0241)	**-0.767** (0.0971)	**0.556** (0.1136)	**1.254** (0.1748)	**1993.01132** -	**2011.85293** -
LACD2	0.5931 (0.7333)	0.0505 (0.0506)	0.6977 (0.3541)	0.5787 (0.1202)	1.2073 (0.1692)	1998.07453 -	2016.91613 -



(a) (b)

Fig. 5. The qq-plot and sample ACF of the residuals for the inter-arrival times. (a) The qq-plot of residuals. (b) ACF of residuals.

[1]In this paper, the term *inter-arrival time*, which is widely used in the computer science community, and the term *duration*, which is widely used in the statistics community, are used interchangeably.

TABLE V
THE p-VALUES OF STATISTICAL TESTS FOR THE RESIDUALS

Test	KS	AD	CM	McLeod-Li	Ljung-Box
p-value	.2312	.2116	.3581	.4045954	.4015984

p-values are all greater than 0.1, meaning that there is no correlation left in the residuals and that the proposed LACD_1 can adequately describe the evolution of the incidents inter-arrival time.

In order to validate the afore-mentioned assumption of the generalized gamma innovations, we report the p-values of the Kolmogorov-Smirnov (KS), Anderson-Darling (AD), and Cramer-von Mises (CM) tests [43] in the left-hand half of Table V. Intuitively, these tests (as reviewed in Appendix VII-D) examine how well the samples fit a theoretical distribution such that a larger p-value indicates a better fit, but using different approaches. The KS test focuses on the largest deviation of the samples from the theoretical distribution, whereas the AD and CM tests consider the overall deviation. We observe that the p-values are .2312, .2116 and .3581, respectively. Therefore, the assumption is validated.

The preceding discussions lead to:

Insight 3: The inter-arrival times of hacking incidents exhibit a significant temporal correlation, and therefore should be modeled by a stochastic process rather than a distribution. Given this, we find that the incidents inter-arrival times can be adequately described by the proposed type-I log-ACD model ($LACD_1$), which implies that the next inter-arrival time is in fact affected by the present one.

B. Modeling the Breach Sizes

In order to model the evolution of the mean of the breach sizes, we propose using the ARMA process, or more specifically ARMA(p, q), where p is the AR order and q is the MA order.the The preceding Insight 2, especially the volatility clustering phenomenon exhibited by the log-transformed breach sizes, suggests that we use a GARCH model to model the volatilities in the breach sizes. An analysis on the residuals suggests that GARCH(1, 1) is sufficient to describe the volatilities in the residuals, which coincides with the conclusion drawn in the literature that higher-order GARCH models are not necessarily better than GARCH(1, 1) [44]. Therefore, we fix the GARCH part as GARCH(1, 1). This leads to the following ARMA-GARCH model:

$$Y_t = \mathrm{E}(Y_t|\mathfrak{F}_{t-1}) + \epsilon_t,$$

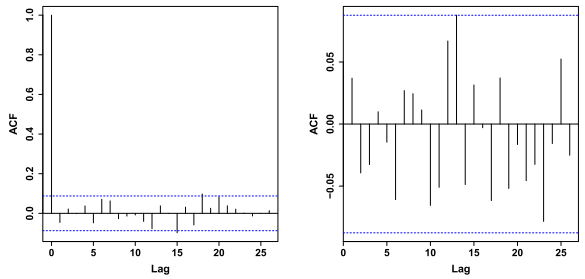
where $\mathrm{E}(\cdot|\cdot)$ is the conditional expectation function, $\mathfrak{F}_{t-1}$ is the historic information up to time $t-1$, and ϵ_t is the innovation of the time series. Since the mean part is modeled as ARMA(p, q), the model can be rewritten as

$$Y_t = \mu + \sum_{k=1}^{p} \phi_k Y_{t-k} + \sum_{l=1}^{q} \theta_l \epsilon_{t-l} + \epsilon_t, \qquad \text{(IV.3)}$$

where $\epsilon_t = \sigma_t Z_t$ with Z_t being the i.i.d. innovations, and the ϕ_k's and the θ_l's are respectively the coefficients of the AR and MA parts. For the standard GARCH(1, 1) model, we have

$$\sigma_t^2 = w + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \qquad \text{(IV.4)}$$

where σ_t^2 is the conditional variance and w is the intercept.



(a) (b)

Fig. 6. Sample ACF of standardized and squared standardized residuals for the log-transformed breach sizes. (a) ACF of standardized residuals. (b) ACF of squared standardized residuals.

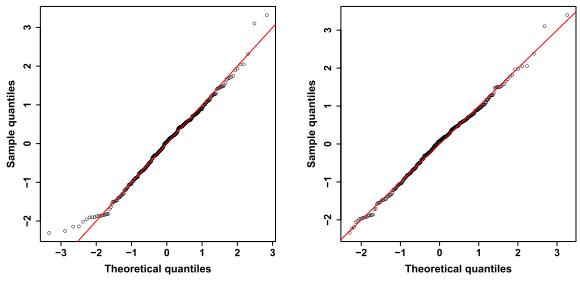
TABLE VI
THE FITTING RESULTS OF THE ARMA(1, 1)-GARCH(1, 1) MODEL FOR THE BREACH SIZES, WHERE THE NUMBERS IN PARENTHESES ARE THE ESTIMATED STANDARD DEVIATIONS

Parameters	μ	ϕ_1	θ_1	ω	α_1	β_1
Estimate	8.680	.859	.709	.277	.071	.893
Standard Deviation	(.238)	(.040)	(.074)	(.231)	(.027)	(.044)

For model selection, we use the AIC criterion to determine the orders of the ARMA models. Note that if ARMA(p, q)-GARCH can successfully accommodate the serial correlations in the conditional mean and the conditional variance, there would be no autocorrelations left in the standardized and squared standardized residuals. When the AIC criterion suggests to select multiple models with similar AIC values, we select the simpler model. The autoregressive p and the moving average order q are allowed to vary between 0 and 5. We find that ARMA(1, 1)-GARCH(1, 1) with normally-distributed innovations is sufficient to remove the serial correlations.

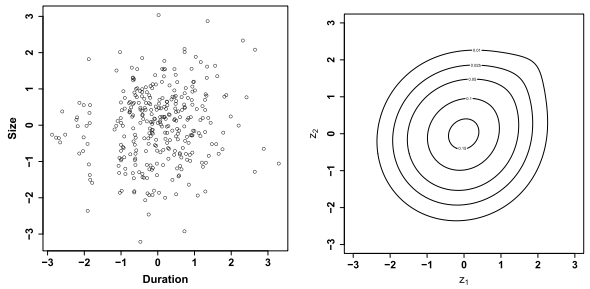
In order to further evaluate the fitting of ARMA(1, 1)-GARCH(1, 1), we plot the sample ACFs for the standardized residuals and the squared standardized residuals in Figure 6. We observe that none of the lags is significant (i.e., the correlations are removed). The p-values of the Ljung-Box tests for both the standardized residuals and the standardized square residuals are very large, namely, .999 and .958, respectively. This means that we cannot reject the null hypothesis that no serial correlations are left in the residuals. Table VI shows the fitting results by ARMA(1, 1)-GARCH(1, 1). We observe that the estimated coefficients for the ARMA and GARCH parts are all statistically significant.

Having observed that ARMA(1, 1)-GARCH(1, 1) can fit the breach sizes overall, we need to know whether or not this model can fit the tails as well. Unfortunately, we observe that normally-distributed innovations fail to capture the tails of the breach sizes because both tails are thick. Therefore, we further consider other distributions for the innovations, including Student-t, generalized error, skewed normal, skewed Student-t, and skewed generalized error distributions. We find that among all these innovation distributions, the skewed Student-t distribution leads to a relatively more accurate fitting. However, as shown by the qq-plot in Figure 7(a), the skewed Student-t still fails to fit the tails. This motivates us to



(a) (b)

Fig. 7. The qq-plots of the residuals of ARMA(1, 1)-GARCH(1, 1) with innovations following different distributions for fitting the log-transformed breach sizes. (a) The qq-plot of the skewed Student-t. (b) The qq-plot of the mixed distribution.



(a) (b)

Fig. 8. Normal score plot and fitted contour plot. (a) Normal scores plot. (b) Gumbel contour plot.

propose an extreme value mixture distribution for describing the innovations.

The Extreme Value Theory (EVT) [32], [45] is a useful tool for modeling the heavy-tail distribution. A popular method is known as the *peaks over threshold* approach (POT). Given a sequence of i.i.d. observations $X_1, \ldots, X_n$, the excesses $X_i - \mu$ of some suitably high threshold μ can be modeled by, under certain mild conditions, the *generalized Pareto distribution* (GPD). The survival function of the GPD

$$\bar{G}_{\xi,\sigma,\mu}(x) = 1 - G_{\xi,\sigma,\mu} = \begin{cases} \left(1 + \xi \dfrac{x-\mu}{\sigma}\right)_+^{-1/\xi}, & \xi \neq 0, \\ \exp\left\{-\dfrac{x-\mu}{\sigma}\right\}, & \xi = 0. \end{cases}$$

where $x \geq \mu$ if $\xi \in \mathbb{R}^+$ and $x \in [\mu, \mu - \sigma/\xi]$ if $\xi \in \mathbb{R}^-$, and ξ and σ are respectively called the *shape* and *scale* parameters. Because Figure 7(a) shows that both tails cannot be modeled by the skewed Student-t distribution, we propose modeling both tails with the GPD and modeling the middle part with the normal distribution. This leads to a mixed extreme value distribution that is used to model the innovations as follows:

$$G_m(x) = \begin{cases} p_l[1 - G(-x|\xi_l, \sigma_l, -\mu_l)], \\ \qquad \text{if } x \leq \mu_l, \\ p_l + (1 - p_l - p_u)\dfrac{\Phi(x|\mu_m, \sigma_m) - \Phi(\mu_l|\mu_m, \sigma_m)}{\Phi(\mu_u|\mu_m, \sigma_m) - \Phi(\mu_l|\mu_m, \sigma_m)}, \\ \qquad \text{if } \mu_l < x < \mu_u, \\ 1 - p_u + p_u G(x|\xi_u, \sigma_u, \mu_u), \\ \qquad \text{if } x \geq \mu_u. \end{cases}$$

where $p_l = P(X \leq \mu_l)$ and $p_u = P(X > \mu_u)$ are the probabilities corresponding to the tails, and μ_m and σ_m are respectively the mean and the standard deviation of the normal distribution. It is worth mentioning that a similar idea has been used to model the impact of the financial crisis on stock and index returns [46], [47].

The estimated parameters for the tail proportions are $(p_l, p_u) = (0.126, 0.098)$, which means that both tails account for about 10% of the observations of GPD. The estimated parameters $\left(\hat{\mu}_m, \hat{\sigma}_m, \hat{\mu}_l, \hat{\sigma}_l, \hat{\xi}_l, \hat{\mu}_u, \hat{\sigma}_u, \hat{\xi}_u\right)$ for the GPD and normal distributions are

$$(-0.002, 0.963, -1.105, 0.877, -0.694, 1.243, 0.471, 0.001).$$

It is interesting to note that the upper tail shape parameter $\xi = .001$ indicates that the upper tail is heavy. The qq-plot in Figure 7(b) indicates that the mixed distribution describes the tails well because all of the points are around the 45-degree line. This leads to:

Insight 4: The log-transformed hacking breach sizes exhibit a significant temporal correlation, and therefore should be modeled by a stochastic process rather than a distribution. Moreover, the log-transformed hacking breach sizes exhibit the volatility clustering phenomenon with possibly extremely large breach sizes. These two properties lead to the development of ARMA$(1, 1)$*-GARCH*$(1, 1)$ *with innovations that follow a mixed extreme value distribution, which can adequately describe the evolution of the log-transformed breach size.*

Note that the ARMA(1, 1) part models the means of the observations and the GARCH(1, 1) part models the large volatility exhibited by the data.

C. Dependence Between Inter-Arrival Times and Breach Sizes

In order to answer the question whether or not there exists dependence between the inter-arrival times and the breach sizes, we propose conducting the *normal score transformation* [35] to the residuals that are obtained after fitting these two time series. For residuals of the LACD_1 fitting, denoted by $e_1, \ldots, e_n$, we use the fitted generalized gamma distribution $G(\cdot|\gamma, k)$ to convert them into empirical normal scores:

$$e_i \to \Phi^{-1}(G(e_i|\gamma, k)), \quad i = 1, \ldots, n,$$

where Φ^{-1} is the inverse of the standard normal distribution. For the residuals of the ARMA(1, 1)-GARCH(1, 1) fitting, we use the estimated mixed extreme value distribution to convert them into empirical normal scores.

Figure 8(a) plots the bivariate normal scores. We observe that large transformed durations are associated with large transformed sizes, implying a positive dependence between the inter-arrival times and the breach sizes. In order to statistically test the dependence, we compute the sample Kendall's τ and Spearman's ρ for the incidents inter-arrival times and the breach sizes, which are 0.07578 and .11515, respectively. The *nonparametric rank* tests [43] for both statistics lead to a p-value of .04313 and .03956, respectively, which are very small. This means that there indeed exists some positive dependence between the inter-arrival times and the breach sizes.

In order to model the bivariate dependence between the incidents inter-arrival times and the breach sizes, we propose using the *Copula* technique [34], [35]. A bivariate copula is a Cumulative Distribution Function (CDF) with uniform marginals on $[0, 1]$. Let X_1 and X_2 be continuous random variables with joint cumulative distribution function

$$F(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2),$$

and univariate marginal distributions F_1 and F_2. A copula C is defined as the joint CDF of the random vector $(F_1(X_1), F_d(X_2))$. From Sklar's theorem [34], [35], the copula C is unique and satisfies

$$F(x_1, x_2) = C(F_1(x_1), F_2(x_2))$$

when the F_i's are all continuous. The corresponding joint density function can be represented as

$$f(x_1, x_2) = c(F_1(x_1), F_2(x_2)) \prod_{i=1}^{2} f_i(x_i),$$

where $c(u_1, u_2)$ is the 2-dimensional copula density function, and f_i is the marginal density function of X_i, $i = 1, 2$.

Suppose at time t, the vector $\mathbf{Z}_t = (Z_{1,t}, Z_{2,t})$ has the following distribution

$$F_z(\mathbf{z}_t; \boldsymbol{\vartheta}, \boldsymbol{\Theta}) = C\left(F(z_{1,t}), G(z_{2,t}); \boldsymbol{\Theta}, \boldsymbol{\vartheta}\right), \qquad \text{(IV.5)}$$

where $\boldsymbol{\Theta}$ denotes the vector of parameters of a copula, $\boldsymbol{\vartheta}$ represents the vector of parameters of the marginal models, and F is the marginal distribution of the residual of the inter-arrival times, and G is the marginal distribution of the residual of the breach sizes. The joint log-likelihood function of the model can be written as

$$\begin{aligned} L(\boldsymbol{\Theta}; \boldsymbol{\vartheta}) = \sum_{t=1}^{n} \Bigg[& \log c\left(F\left(\frac{d_t}{\Psi_t}\right), G\left(\frac{y_t - \mu_t}{\sigma_t}\right); \boldsymbol{\vartheta}, \boldsymbol{\Theta}\right) \\ & - \log(\sigma_t) - \log(\Psi_t) + \log\left(g\left(\frac{y_t - \mu_t}{\sigma_t}; \boldsymbol{\vartheta}\right)\right) \\ & + \log\left(f\left(\frac{d_t}{\Psi_t}\right); \boldsymbol{\vartheta}\right)\Bigg], \end{aligned}$$

where $c(\cdot)$ is the copula density of $C(\cdot)$, $\mu_t = \mathrm{E}(Y_t|\mathfrak{F}_{t-1})$, $f(\cdot)$ is the density function of $Z_{1,t}$, and $g(\cdot)$ is the density function of $Z_{2,t}$.

A popular method for estimating the parameters of a joint model is the Inference Function of Margins method [48]. This method has two steps: (i) estimate the parameters of the marginal stochastic models; and (ii) estimate the parameters of the copula by fixing the parameters obtained at step (i). Since we have identified the stochastic models for the inter-arrival times and the breach sizes, in what follows we discuss how to model the bivariate dependence.

There are many bivariate copulas [34], [35]. We consider a range of them by using the state-of-art R package VineCopula, and Table VII describes the fitting results of these copulas. We observe that the Gumbel copula has the smallest AIC and BIC, which confirms what is hinted by Figure 8(a), namely that there exists a right-tail dependence between the inter-arrival times and the breach sizes. Figure 8(b) plots the fitted Gumbel contour, indicating an accurate fitting.

TABLE VII
DEPENDENCE MODEL FITTING

Model	Log-likelihood	AIC	BIC
Gumbel	3.63	**-5.27**	**-1.5**
Tawn type 1	4.36	-4.72	2.82
BB8	3.76	-3.52	4.02
Survival Clayton	2.72	-3.45	0.32
Gaussian	2.64	-3.27	0.5
Joe	3.34	-4.67	-0.91
BB6	3.63	-3.27	4.27

In order to further examine the dependence fitting of Gumbel copula, we use two goodness-of-fit tests: (i) the White test [49], [50], which leads to a test statistic of .0648 and a p-value of 0.2626 (meaning that the dependence can be modeled by the Gumbel copula); (ii) The Cramer-von Mises statistic [51], [52], which leads to a test statistic of .1379 and a p-value of 0.1212 (meaning that the dependence can be modeled by the Gumbel copula). Since the p-values are large for both tests, we conclude that the Gumbel copula can adequately describe the dependence between the inter-arrival times and the breach sizes.

Insight 5: There exists a statistical positive dependence between the hacking breach incidents inter-arrival times and the breach sizes. The cybersecurity meaning of the dependence is that if there is a long period of time during which there are no hacking breach incidents, then it is more likely to have a large hacking breach when an incident occurs.

The situation of cyber hacking breaches reflects the outcome of the cyber attack-defense interactions (e.g., whether or not the attack tools can successfully evade the defense tools). Although the particular phenomenon mentioned above can happen under many different scenarios and precisely pinning down of its cause is beyond the scope of the present paper (simply because of the lack of various kinds of supporting data), one possibility is the following: When the attack tools are no longer effective from the attacker's point of view, the attackers may need to take a longer period of time to develop new attack tools for successfully breaching data.

V. PREDICTION

Having showed how to fit the inter-arrival times and the breach sizes, now we investigate how to predict them.

A. Prediction Evaluation Metric

Let us recall the Value-at-Risk (VaR) [53] metric. For a random variable X_t of interest, the VaR at level α, where $0 < \alpha < 1$, is defined as

$$\mathrm{VaR}_\alpha(t) = \inf\{l : P(X_t \leq l) \geq \alpha\}.$$

For example, $\mathrm{VaR}_{.95}(t)$ means that there is only a 5% probability that the observed value is greater than the predicted value $\mathrm{VaR}_{.95}(t)$. An observed value greater than the predicted $\mathrm{VaR}_\alpha(t)$ is called a *violation*, indicating inaccurate prediction. In order to evaluate the prediction accuracy of the VaR

Algorithm 1 Algorithm for Predicting the VaR_α's of the Hacking Incidents Inter-Arrival Times and the Breach Sizes *Separately*

Input: Historical incidents inter-arrival times and breach sizes, denoted by $\{(d_{t_i}, y_{t_i})\}_{i=1,\ldots,m+n}$, where an in-sample $\{(d_{t_i}, y_{t_i})\}_{i=1,\ldots,m}$ as mentioned above was used for fitting and an out-of-sample $\{(d_{t_i}, y_{t_i})\}_{i=m+1,\ldots,n}$ is used for evaluation prediction accuracy; α level.

1: **for** $i = m+1, \cdots, n$ **do**
2: Estimate the LACD_1 model of the incidents inter-arrival times based on $\{d_s | s = 1, \ldots, i-1\}$, and predict the conditional mean $\Psi_i = \exp(\omega + a_1 \log(\epsilon_{i-1}) + b_1 \log(\Psi_{i-1}))$;
3: Estimate the ARMA-GARCH of log-transformed size, and predict the next mean $\hat{\mu}_i$ and standard error $\hat{\sigma}_i$;
4: Select a suitable Copula using the bivariate residuals from the previous models based on AIC;
5: Based on the estimated copula, simulate 10000 2-dimensional copula samples $\left(u_{1,i}^{(k)}, u_{2,i}^{(k)}\right)$, $k = 1, \ldots, 10000$;
6: For the incidents inter-arrival times, convert the simulated dependent samples $u_{1,i}^{(k)}$'s into the $z_{1,i}^{(k)}$'s by using the inverse of the estimated generalized gamma distribution, $k = 1, \ldots, 10000$;
7: For the breach sizes, convert the simulated dependent samples $u_{2,i}^{(k)}$'s into the $z_{2,i}^{(k)}$'s by using the inverse of the estimated mixed extreme value distribution, $k = 1, \ldots, 10000$;
8: Compute the predicted 10000 2-dimensional breach data $\left(d_i^{(k)}, y_i^{(k)}\right)$,$k = 1, \ldots, 10000$ based on Eq. (IV.1) and (IV.3), respectively;
9: Compute the $\text{VaR}_{\alpha,d}(i)$ for the incidents inter-arrival times and $\text{VaR}_{\alpha,y}(i)$ for the log-transformed breach sizes based on the simulated breach data.
10: **if** $d_i^{(k)} > \text{VaR}_{\alpha,d}(i)$ **then**
11: A violation to the incidents inter-arrival time occurs;
12: **end if**
13: **if** $y_i^{(k)} > \text{VaR}_{\alpha,y}(i)$; **then**
14: A violation to the breach size occurs;
15: **end if**
16: **end for**

Output: Numbers of violations in inter-arrival times and breach sizes.

values, we use the following three popular tests [54]. The first test is the unconditional coverage test, denoted by LR_{uc}, which evaluates whether or not the fraction of violations is significantly different from the model's violations. The second test is the conditional coverage test, denoted by LR_{cc}, which is a joint likelihood ratio test for the independence of violations and unconditional coverage. The third test is the dynamic quantile test (DQ) [55], which is based on the sequence of 'hit' variables.

B. Algorithm for Separate Prediction and Results

We use Algorithm 1 to perform the recursive rolling prediction for the inter-arrival time and the breach sizes. Because we use rolling prediction, meaning that training data grows as the prediction operation moves forward, newer training data needs to be re-fitted, possibly needing different copula models. As such, we need to consider more dependence structure. This explains why we need to re-select the copula structure, which can fit the newly updated training data better, via the criterion of AIC (see Step 4 of Algorithm 1).

TABLE VIII
VAR TESTS OF PREDICTED INTER-ARRIVAL TIMES AND BREACH SIZES AT LEVELS $\alpha = .90, .92, .95$

	α	Ob.	Exp	LRuc	LRcc	DQ
inter-arrival time	.90	26	28	.6871	.8522	.9523
inter-arrival time	.92	21	22	.7554	.5157	.6931
inter-arrival time	.95	12	14	.5743	.4979	.4352
breach size	.90	31	28	.5561	.8099	.9996
breach size	.92	27	22	.2336	.4881	.9999
breach size	.95	20	14	.1210	.2673	.9999

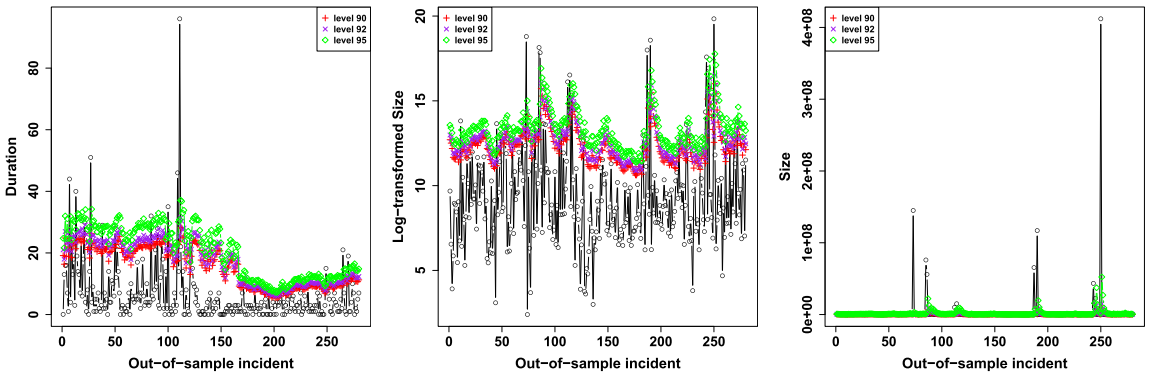
Table VIII reports the prediction results. We observe that the prediction models pass all of the tests at the .1 significant level. In particular, the models can predict the future inter-arrival times for all of the α's levels. For the breach sizes, at level $\alpha = .90$, the model predictions have 28 violations, while the number of violations from the observed values is 31, which is fairly close to each other. For $\alpha = .95$, the number of violations from the observed values is 20, while the model's expected number of violations is 14. This indicates that the models for predicting the future breach sizes are somewhat conservative.

Figure 9 plots the prediction results for the 280 out-of-samples. Figure 9(a) plots the prediction results for the incidents inter-arrival times. Figure 9(c) plots of the original breach sizes, but it is hard to look into visually. For a better visualization effect, we plot in Figure 9(b) the log-transformed breach sizes. We observe from Figure 9(c) that for the breach sizes, there are several extreme large values, which are far from the predicted $\text{VaR}_{.95}$'s. This means that the prediction missed some of the extremely large breaches, the prediction of which is left as an open problem.

In conclusion, the proposed models can effectively predict the VaR's of both the incidents inter-arrival time and the breach size, because they both pass the three statistical tests. However, there are several extremely large inter-arrival times and extremely large breach sizes that are far above the predicted $\text{VaR}_{.95}$'s, meaning that the proposed models may not be able to precisely predict the exact values of the extremely large inter-arrival times or the extremely large breach sizes. Nevertheless, as shown in Section V-C below, our models can predict the joint probabilities that an incident of a certain *magnitude* of breach size will occur during a future period of time.

C. Algorithm for Joint Prediction and Results

In practice, it is important to know the joint probability that the next breach incident of a particular size happens at a particular time (i.e., with a particular inter-arrival time). For this purpose, we consider the 10000 values predicted



(a) (b) (c)

Fig. 9. Predicted inter-arrival times and breach sizes, where black-colored circles represent the observed values. (a) Incidents inter-arrival times. (b) Log-transformed breach sizes. (c) Breach sizes (prior to the transformation).

TABLE IX

PREDICTED JOINT PROBABILITIES OF INCIDENTS INTER-ARRIVAL TIMES AND BREACH SIZES, WHERE "PROB." IS THE PROBABILITY OF BREACH SIZE A CERTAIN PREDICTED y_t OCCURRING WITH THE NEXT TIME $d_t \in (0, \infty)$

Inter-arrival time / Breach size	Prob.	$d_t \in (30, \infty)$	$d_t \in (14, 30]$	$d_t \in (7, 14]$	$d_t \in (1, 7]$	$d_t \in (0, 1]$
	Copula model					
$y_t \in (1 \times 10^6, \infty)$	0.0460	0.0002	0.0042	0.0084	0.0233	0.0099
$y_t \in (5 \times 10^5, 1 \times 10^6]$	0.0217	0.0001	0.0013	0.0038	0.0129	0.0036
$y_t \in (1 \times 10^5, 5 \times 10^5]$	0.1107	0.0002	0.0075	0.0200	0.0530	0.0300
$y_t \in (5 \times 10^4, 1 \times 10^5]$	0.0890	0.0005	0.0055	0.0163	0.0463	0.0204
$y_t \in (1 \times 10^4, 5 \times 10^4]$	0.2544	0.0005	0.0166	0.0409	0.1240	0.0724
$y_t \in (5 \times 10^3, 1 \times 10^4]$	0.1156	0.0003	0.0075	0.0178	0.0551	0.0349
$y_t \in (1 \times 10^3, 5 \times 10^3]$	0.2089	0.0005	0.0110	0.0305	0.1035	0.0634
$y_t \in [1, 1 \times 10^3]$	0.1537	0.0001	0.0068	0.0212	0.0732	0.0524
Total	1	0.0024	0.0604	0.1589	0.4913	0.2870
	Benchmark model					
$y_t \in (1 \times 10^6, \infty)$	0.0339	0.0002	0.0018	0.0064	0.0176	0.0079
$y_t \in (5 \times 10^5, 1 \times 10^6]$	0.0224	0.0002	0.0019	0.0033	0.0102	0.0068
$y_t \in (1 \times 10^5, 5 \times 10^5]$	0.1112	0.0003	0.0070	0.0160	0.0560	0.0319
$y_t \in (5 \times 10^4, 1 \times 10^5]$	0.0913	0.0002	0.0061	0.0163	0.0425	0.0262
$y_t \in (1 \times 10^4, 5 \times 10^4]$	0.2568	0.0003	0.0165	0.0439	0.1260	0.0701
$y_t \in (5 \times 10^3, 1 \times 10^4]$	0.1121	0.0003	0.0070	0.0160	0.0554	0.0334
$y_t \in (1 \times 10^3, 5 \times 10^3]$	0.2170	0.0009	0.0116	0.0356	0.1066	0.0623
$y_t \in [1, 1 \times 10^3]$	0.1553	0.0007	0.0102	0.0261	0.0779	0.0404
Total	1	0.0031	0.0621	0.1636	0.4922	0.2790

by Algorithm 1. Specifically, we consider several combinations of (d_i, y_{t_i}), where $d_i = t_i - t_{i-1}$ and y_{t_i} is the breach size at time t_i for $i = 1, \ldots, n$ as mentioned above.

We divide the predicted inter-arrival time of the next breach incident into the following time intervals: (i) longer than one month or $d_t \in (30, \infty)$; (ii) in between two weeks and one month or $d_t \in (14, 30]$; (iii) in between one and two weeks $d_t \in (7, 14]$; (iv) in between one day and one week $d_t \in (1, 7]$; (v) within one day $d_t \in (0, 1]$. Similarly, we divide the predicted breach size of the next breach incident into the following size intervals: (i) greater than one million records or $y_t \in (1 \times 10^6, \infty)$, indicating a large breach; (ii) $y_t \in (5 \times 10^5, 1 \times 10^6]$; (iii) $y_t \in (1 \times 10^5, 5 \times 10^5]$; (iv) $y_t \in (5 \times 10^4, 1 \times 10^5]$; (v) $y_t \in (1 \times 10^4, 5 \times 10^4]$; (vi) $y_t \in (5 \times 10^3, 1 \times 10^4]$; (vii) $y_t \in (1 \times 10^3, 5 \times 10^3]$; (viii) smaller than 1000 or $y_t \in [1, 1 \times 10^3]$, indicating a small breach. We use the models mentioned above to fit these bivariate observations, and predict the joint event by using Algorithm 1 (steps 2-8).

Table IX describes the predicted probabilities of joint events (d_t, y_t) using the copula model, as well as the predicted joint probabilities by using the benchmark model, which makes the independence assumption between the incidents inter-arrival times and the breach sizes. We observe that these probabilities are different from that of the benchmark model. For example, the probability of data breach is .0460 for breach sizes exceeding one million (i.e., severe breach incidents), namely $y_t \in (1 \times 10^6, \infty)$, while the probability based on the benchmark model is only .0339. Moreover, when we look at the joint event of inter-arrival time $d_t \in (0, 7)$ and breach size $y_t \in (1 \times 10^6, \infty)$, the copula model predicts the probability as .0332; whereas, the benchmark model predicts the probability as .0255. This means that the benchmark model underestimates the severity of data breach incidents.

We further observe that both models predict that there will be a breach incident occurring within a month, where the copula model predicts the probability of this incident being .9976, and the benchmark model predicts this probability being .9969. This indicates that almost certainly a data breach incident will happen within a month. Further, the copula model predicts a probability of .7783 that a breach incident will occur within a week, while the benchmark model predicts

this probability as .7712. This means that there is a high chance that a data breach incident will happen within a week. When we reexamine the database by PRC, there was a data breach reported on April 12, 2017 with 1.3 million records breached. Note that our model uses the data ending on t_n equals April 7, 2017, meaning that the incident happened during a week as predicted by our model.

The other interesting discovery is that the model predicted the following: the probability that a new incident will occur within one day (i.e., April 8, 2017) with a probability of 0.287. After looking into the original the dataset, we find no incident that was reported on April 8, 2017. Therefore, a cyber incident may not be recorded with chance 28.7%. Moreover, the prediction result says that if there is indeed an incident that was not recorded, the probability that the breach size of the incident exceeds 500,000 is very low (0.047); with probability 0.7774, the breach size was less than 50,000.

By summarizing the preceding discussion, we draw:

Insight 6: The proposed approach can accurately predict the joint probability that the next hacking breach incident occurs during a particular period of time and the corresponding breach size falls into a particular interval (i.e., the probability that an incident of a certain magnitude of breach size will occur within a certain period of time).

In practice, if one is interested in predicting the particular breach size at a particular future point in time, the former method should be used, with the "caveat" that the predicted value has a no-more-than 5% chance of being smaller than the actual value that will be observed. If one is interested in predicting the joint probability that a breach incident with a certain magnitude of breach size during a certain future period of time, the latter method should be used. This kind of prediction capability is, like weather forecasting (e.g., a hurricane of a certain degree will occur within the next 5 days), useful because cyber defenders can dynamically adjust their defense posture to mitigate the damage, ranging from temporarily shutting down unnecessary services (if applicable) to allocating additional resources in examining network traffic (e.g., expensive but effective deep packet inspections or large-scale data correlation analyses). Moreover, the prediction model might help estimate the budget in a defense strategy planning. This is important because the effort spent to defend an enterprise against an attack (e.g. the amount of cost incurred by a certain defense) depends on the likelihood of an attack to happen and its severeness (i.e., quantitative risk management). For instance, when the model predicts that a huge data breach is unlikely to happen, the defenses for that attack can be less sophisticated (ratio cost-effectiveness); when the model predicts that a huge data breach is likely to happen, the defender can set up more delicate defenses (e.g., honeypots and more accurate audit systems). We believe that these types of predictive-defense (i.e., dynamic defense enabled by prediction capability) are an important topic for future research, as analogously justified by the usefulness of weather forecasting in the physical world.

VI. Trend Analysis

In this section we present both qualitative and quantitative trend analyses on the hacking breach incidents based on the models presented above. For this purpose, we decompose the data into two parts: the trend part and the random (or noise) part. In general, the trend part refers to the pattern that is exhibited by the data and can be modeled via the technical/statistical analysis (e.g. linear, nonlinear, and cyclic/seasonal trends), and the random part refers to the remainder of the data after removing the trend part [38].

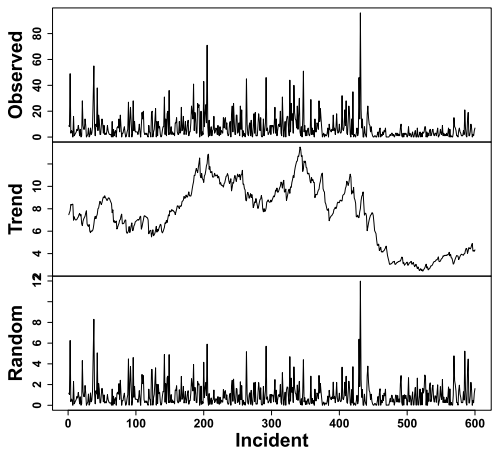


Fig. 10. Using the LACD_1 model to decompose the hacking breach incidents inter-arrival times into a trend part and a random part.

A. Qualitative Trend Analysis

1) Qualitative Trend Analysis of the Hacking Breach Incidents Inter-Arrival Times: In Section IV-A, we showed that the LACD_1 model can describe the breach incidents inter-arrival times. The trend is formally defined as:

$$\log(\Psi_i) = \omega + a_1 \log(\epsilon_{i-1}) + b_1 \log(\Psi_{i-1}),$$

namely the LACD_1 model, and the random part is defined as ϵ_i, which is modeled by the generalized gamma distribution in Eq. (IV.2). The estimated parameters of which are

$$(\omega, a_1, b_1, k, \gamma) = (3.825, 0.058, -0.767, 0.556, 1.254),$$

and the estimated standard deviations of these parameters are respectively (0.2254, 0.0241, 0.0971, 0.1136, 0.1748). We observe that all these parameters are significant.

Figure 10 plots the decomposed time series of the inter-arrival times: the top-panel corresponds to the observed data; the middle-panel corresponds to the trend; and the bottom-panel corresponds to the random noise. We observe from the middle-panel that the inter-arrival time shows a decreasing trend in the recent years (say, after the 415th incident occurring on 12/18/2014), and then is followed by a slightly increasing trend (say, after the 521st incident occurring on 06/14/2016). This implies that hacking breach incidents happen more frequently prior to 06/14/2016 (because the incident inter-arrival times are shorter) and less frequently after 06/14/2016 (because the incident inter-arrival times are longer).

In order to further study the trend of the inter-arrival times, we plot the estimated $\text{VaR}_{.9}$ corresponding to the time interval between 12/18/2014 and 04/12/2017 in Figure 11. We observe that the VaR first shows a decreasing trend and then a slightly increasing pattern. This indicates that the hacking breach incidents first become worse and then become somewhat less frequent from the perspective of the inter-arrival time.

TABLE X

QUANTITATIVE TREND ANALYSIS STATISTICS OF HACKING BREACH INCIDENTS, WHERE 'SD' STANDS FOR STANDARD DEVIATION

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
breach-size GR													
Min	-0.9982	-0.9994	-0.9993	-0.9997	-0.9999	-0.9995	-0.9998	-0.9994	-0.9980	-1.0000	-0.9984	-1.0000	-0.9943
Mean	39.4934	226.5891	66.3486	58.4757	52.1951	274.1949	197.2350	67.1771	1144.7224	307.1408	20.6467	3917.1173	39.7667
Median	-0.1294	-0.4275	-0.4113	-0.1250	3.3333	-0.3693	2.2543	0.1538	-0.2633	-0.2878	2.0172	0.2699	-0.3092
Max	999.0000	4922.0769	1317.1818	1319.0000	821.2222	6665.6667	4863.8649	1179.6375	38635.3636	4544.4545	175.5147	411999.0000	394.8333
SD	172.5264	826.8532	271.3574	218.6454	163.7172	1220.3355	767.3975	229.6633	6624.4522	940.2390	41.1997	40014.0813	109.4008
Inter-arrival time GR													
Min	-0.9388	-0.8684	-0.9355	-0.9444	-0.9500	-0.9091	-0.9474	-0.9286	-0.9310	-0.8929	-0.9429	-0.9000	-0.9474
Mean	1.6624	0.6566	1.2135	1.2014	1.7443	1.3989	1.7605	3.4078	1.0936	0.5140	2.1984	0.5854	1.3366
Median	0.4167	0.0000	0.0000	0.3000	0.0000	0.2000	0.0625	0.0417	0.2000	-0.1667	-0.3542	0.0000	0.0000
Max	27.0000	8.3333	17.0000	17.0000	20.5000	23.0000	14.0000	43.0000	8.6667	7.0000	16.5000	9.5000	12.0000
SD	4.7815	1.8340	3.4678	3.3193	4.5958	3.9847	3.8778	9.9052	2.3000	1.5740	5.2795	1.6123	3.5511
AGRT													
Min	-0.8846	-0.4997	-0.8200	-0.9800	-0.9949	-0.9975	-0.9997	-0.9758	-0.9667	-0.9955	-0.9943	-0.9959	-0.9700
Mean	2.1217	29.5880	8.7768	5.8103	7.2840	59.4998	128.7676	3.8688	285.1684	61.4814	5.7890	500.1983	33.1510
Median	-0.0109	-0.0428	-0.0424	-0.0264	0.3003	-0.0406	0.2518	0.0129	-0.0318	-0.0360	0.1390	0.0924	-0.0406
Max	35.6786	615.2596	252.9646	69.4211	74.1445	2221.8889	4863.8649	65.5354	9658.8409	999.6667	79.8147	51499.8750	394.8333
SD	7.2163	101.9517	38.9521	15.7273	18.6590	339.1464	731.9553	13.1144	1656.2877	195.4920	16.2583	5001.0840	108.8875
CGRT													
Min	-0.8846	-0.9753	-0.8200	-0.9800	-0.9949	-0.9975	-0.9997	-0.9758	-0.9667	-0.9997	-0.9943	-0.9959	-0.9700
Mean	0.2467	1.0643	0.5223	0.1876	3.1194	0.9586	120.9637	-0.0636	0.6061	0.9333	3.8165	6.7059	32.0522
Median	-0.0124	-0.0735	-0.0680	-0.0487	0.0975	-0.0620	0.1185	0.0096	-0.0644	-0.0520	0.0605	0.0808	-0.0747
Max	6.3786	20.5849	7.5551	2.7505	74.1445	17.8207	4863.8649	0.8894	13.0200	13.4225	79.8147	365.1267	394.8333
SD	1.3998	4.0520	1.5995	0.8900	14.8320	3.6782	732.5735	0.4832	2.6462	2.8091	15.8333	38.2205	109.1836

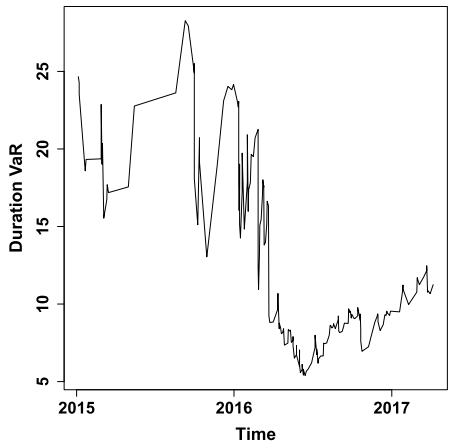


Fig. 11. The estimated $\text{VaR}_{.9}$'s of the hacking breach incidents inter-arrival times based on the LACD_1 model.

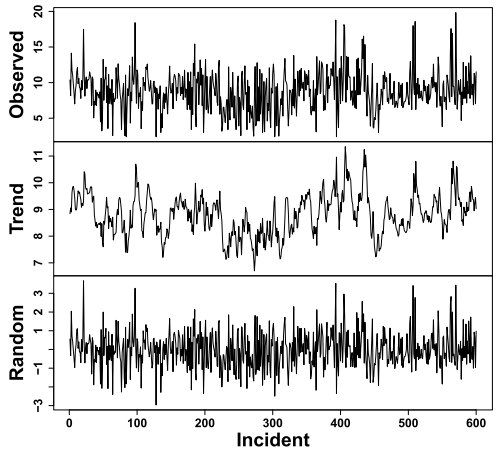


Fig. 12. Using the ARMA-GARCH model to decompose the log-transformed breach sizes into a trend part and a random part.

This finding is different from the conclusion drawn in [9], which was based on a super dataset in terms of the incident types (i.e., *negligent breaches* and *malicious breaching* as we will discuss in Section I-B); whereas, the present study focuses on hacking breach incidents only (i.e., a proper sub-type of the *malicious breaches* type analyzed in [9]).

2) Qualitative Trend Analysis of the Hacking Breach Sizes: In Section IV-B, we used the ARMA-GARCH model with innovations that follow the mixed extreme value distribution to describe the log-transformed breach sizes. Figure 12 plots the decomposition of the time series using this model. The trend is defined as

$$Y_t = \mu + \phi_1 Y_{t-1} + \theta_1 \epsilon_{t-1},$$

and the random part is defined as ϵ_t, which is modeled by the GARCH(1, 1) model described in Eq. (IV.4). We observe that although the breach sizes vary over time, there is no clear trend. This conclusion coincides with what was concluded in [9], which is drawn from, as mentioned above, a proper super set of the dataset we analyze.

B. Quantitative Trend Analysis

In order to quantify the trend, we propose using two metrics to characterize the *growth* of hacking breach incidents. Recall that $\{(t_i, y_{t_i})\}_{i=1,\ldots,n}$ is the sequence of breach incidents occurring at time t_i with a breach size y_{t_i}. Inspired by the growth rate analysis in economics [56], we propose:

- Growth Rate (GR): We define the breach-size GR as

$$\mathsf{GR}_i = \frac{y_{t_{i+1}} - y_{t_i}}{y_{t_i}}.$$

  Inter-arrival times GR can be defined similarly.
- Average Growth Rate over Time (AGRT): We define the AGRT as

$$\mathsf{AGRT}_i = \frac{1}{d_{i+1}} \frac{y_{t_{i+1}} - y_{t_i}}{y_{t_i}}.$$

- Compound Growth Rate over Time (CGRT): We define the CGRT as

$$\mathsf{CGRT}_i = \left(\frac{y_{t_{i+1}}}{y_{t_i}}\right)^{1/d_{i+1}} - 1.$$

Note that AGRT represents the percentage change of the breach size over time, and CGRT describes the rate at which the breach size would grow.

Table X summarizes the results of the quantitative trend analysis. For the breach-size GR, we observe that the means of the GR are all positive, meaning that the breach size becomes increasingly larger each year. Note that the means of the GR

are largely affected by the extreme GR. For example, for year 2016, we have the maximum GR 411, 999, which leads to a very large mean GR (i.e., 3,917.1173). In terms of the medians, we observe that from 2005 to 2008, the GRs are negative, meaning that the breach sizes decrease during these years. The negative GRs of breach sizes are also observed for years 2010, 2013 and 2014. For years 2015 and 2016, we observe positive GRs, 2.0172 and 0.2699, meaning that the breach size increases for these two years. For year 2017, we have a negative median GR (i.e., −0.3092) until April 7, 2017. It is worth mentioning that for years 2010, 2013, and 2016, we have very large standard deviations, which indicate that there exist extreme breach sizes during these years.

For the inter-arrival time GR, we observe that the median GR for each year is relatively small. In particular, we observe that the median is 0 for years 2007, 2007, 2009, 2016, and 2017, meaning that during these years, the breach inter-arrival times are relatively stable. We also observe that for years 2014 and 2015, the medians of the inter-arrival time are negative, meaning that the inter-arrival time decreases for these years. We also note that since year 2012 (except for year 2015), the standard deviations of the GRs of the inter-arrival time are relatively small (smaller than 3.6). We conclude that hacking breach incidents inter-arrival time decreases in recent years. This deepens the qualitative trend analysis in the previous section.

The AGRT and CGRT metrics consider both the breach size and the inter-arrival time. We observe that the means of the AGRT are all positive, meaning that the breach size increases on average. In terms of the median, we observe that the AGRTs of years 2013 and 2014 are negative. Compared to the GRs of these two years, we observe that the absolute values of the AGRTs are smaller, namely, 0.0318 and 0.0360 for the AGRTs versus 0.2633 and 0.2878 for the GRs, respectively. This can be explained by the evolution of the inter-arrival times. Based on AGRT, we conclude that although the breach size turns to be smaller (negative growth) in years 2013 and 2014, it becomes larger (positive growth) in years 2015 and 2016, and becomes smaller at the beginning of year 2017. A similar conclusion can be drawn for the CGRT metric. The median value 0.0808 of CGRT in year 2016 can be interpreted as the median daily growth rate of 0.0808 for year 2016.

By summarizing the preceding qualitative and quantitative trend analysis, we draw:

Insight 7: The situation of hacking breach incidents are getting worse in terms of their frequency, but appear to be stabilizing in terms of their breach sizes, meaning that more devastating breach incidents are unlikely in the future.

VII. Conclusion

We analyzed a hacking breach dataset from the points of view of the incidents inter-arrival time and the breach size, and showed that they both should be modeled by stochastic processes rather than distributions. The statistical models developed in this paper show satisfactory fitting and prediction accuracies. In particular, we propose using a copula-based approach to predict the joint probability that an incident with a certain magnitude of breach size will occur during a future period of time. Statistical tests show that the methodologies proposed in this paper are better than those which are presented in the literature, because the latter ignored both the temporal correlations and the dependence between the incidents inter-arrival times and the breach sizes. We conducted qualitative and quantitative analyses to draw further insights. We drew a set of cybersecurity insights, including that the threat of cyber hacking breach incidents is indeed getting worse in terms of their frequency, but not the magnitude of their damage. The methodology presented in this paper can be adopted or adapted to analyze datasets of a similar nature.

There are many open problems that are left for future research. For example, it is both interesting and challenging to investigate how to predict the extremely large values and how to deal with missing data (i.e., breach incidents that are not reported). It is also worthwhile to estimate the exact occurring times of breach incidents. Finally, more research needs to be conducted towards understanding the predictability of breach incidents (i.e., the upper bound of prediction accuracy [24]).

Appendix

A. ACF and PACF

ACF and PACF [36] are two important tools for examining temporal correlations. Consider a sequence of samples $\{Y_1, \ldots, Y_n\}$. The sample ACF is defined as

$$r_k = \frac{\sum_{t=k+1}^{n} \left(Y_t - \bar{Y}\right)\left(Y_{t-k} - \bar{Y}\right)}{\sum_{t=k+1}^{n} \left(Y_t - \bar{Y}\right)^2}, \quad k = 1, \ldots, n-1,$$

where $\bar{Y} = \sum_{t=1}^{n} Y_t/n$ is the sample mean. The PACF is defined as a conditional correlation of two variables given the information of the other variables. Specifically, the PACF of (Y_t, Y_{t-k}) is the autocorrelation between Y_t and Y_{t-k} after removing any linear dependence on $Y_{t+1}, Y_{t+2}, \ldots, Y_{t-k+1}$; see [36] for more details.

B. AIC and BIC

AIC and BIC are the most commonly used criteria in the model selection in the statistics [36], [37], [53]. AIC is meant to balance the goodness-of-fit and the penalty for model complexity (the smaller the AIC value, the better the model). Specifically,

$$\text{AIC} = -2\log(\text{MLE}) + 2\ k,$$

where MLE is the likelihood associated to the fitted model and measures the goodness-of-fit, and k is the number of estimated parameters and measures the model complexity. Similarly, the smaller the BIC value, the better the model. Specifically,

$$\text{BIC} = -2\log(\text{MLE}) + k\log(n),$$

where n is the sample size. BIC penalizes complex models more heavily than AIC, thus favoring simpler models.

C. Ljung-Box and McLeod-Li Tests

The Ljung-Box test consider a group of ACFs of a time series [37], [57]. The null hypotheses is

$$\text{H}_0 : \text{The time series are independent.}$$

and the alternative is

$$\text{H}_a : \text{The time series are not independent.}$$

The Ljung-Box test statistic is defined as

$$Q = n(n+2)\left(\frac{\hat{r}_1^2}{n-1} + \cdots + \frac{\hat{r}_k^2}{n-k}\right),$$

where $\hat{r}_i$ is the estimated correlation coefficient at lag i. We reject the null hypothesis if $Q > \chi^2_{1-\alpha,k}$ where $\chi^2_{1-\alpha,k}$ is the αth quantile of the chi-squared distribution with k degrees of freedom.

The McLeod-Li test is similarly defined but it tests whether the first m autocorrelations of squared data are zero using the Ljung-Box test [31], [57].

D. Goodness-of-Fit Test Statistics

The goodness-of-fit of a distribution describes how well the distribution fits a set of samples. Three commonly used test statistics are: the Kolmogorov-Smirnov (KS) test, the Anderson-Darling (AD) test, and the Cramér-von Mises (CM) test [58], [59]. Specifically, let $X_1, \ldots, X_n$ be independent and identical random variables with distribution F. The empirical distribution F_n is defined as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}(X_i \leq x),$$

where $\mathrm{I}(X_i \leq x)$ is the indicator function:

$$\mathrm{I}(X_i \leq x) = \begin{cases} 1, & X_i \leq x, \\ 0, & o/w. \end{cases}$$

The KS, CM, and AD test statistics are defined as:

$$\mathrm{KS} = \sqrt{n}\sup_x |F_n(x) - F(x)|,$$

$$\mathrm{CM} = n\int (F_n(x) - F(x))^2 \, dF(x),$$

$$\mathrm{AD} = n\int (F_n(x) - F(x))^2 \, w(x) dF(x),$$

where $w(x) = [F(x)(1-F(x))]^{-1}$.

ACKNOWLEDGMENT

The authors thank the reviewers for their constructive comments that helped improve the paper. In Section V, they incorporated some insightful comments of one reviewer on how to connect the prediction models to real-world cyber defense quantitative risk management.

REFERENCES

[1] P. R. Clearinghouse. *Privacy Rights Clearinghouse's Chronology of Data Breaches*. Accessed: Nov. 2017. [Online]. Available: https://www.privacyrights.org/data-breaches

[2] ITR Center. *Data Breaches Increase 40 Percent in 2016, Finds New Report From Identity Theft Resource Center and CyberScout*. Accessed: Nov. 2017. [Online]. Available: http://www.idtheftcenter.org/2016databreaches.html

[3] C. R. Center. *Cybersecurity Incidents*. Accessed: Nov. 2017. [Online]. Available: https://www.opm.gov/cybersecurity/cybersecurity-incidents

[4] *IBM Security*. Accessed: Nov. 2017. [Online]. Available: https://www.ibm.com/security/data-breach/index.html

[5] NetDiligence. *The 2016 Cyber Claims Study*. Accessed: Nov. 2017. [Online]. Available: https://netdiligence.com/wp-content/uploads/2016/10/P02_NetDiligence-2016-Cyber-Claims-Study-ONLINE.pdf

[6] M. Eling and W. Schnell, "What do we know about cyber risk and cyber risk insurance?" *J. Risk Finance*, vol. 17, no. 5, pp. 474–491, 2016.

[7] T. Maillart and D. Sornette, "Heavy-tailed distribution of cyber-risks," *Eur. Phys. J. B*, vol. 75, no. 3, pp. 357–364, 2010.

[8] R. B. Security. *Datalossdb*. Accessed: Nov. 2017. [Online]. Available: https://blog.datalossdb.org

[9] B. Edwards, S. Hofmeyr, and S. Forrest, "Hype and heavy tails: A closer look at data breaches," *J. Cybersecur.*, vol. 2, no. 1, pp. 3–14, 2016.

[10] S. Wheatley, T. Maillart, and D. Sornette, "The extreme risk of personal data breaches and the erosion of privacy," *Eur. Phys. J. B*, vol. 89, no. 1, p. 7, 2016.

[11] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling Extremal Events: For Insurance and Finance*, vol. 33. Berlin, Germany: Springer-Verlag, 2013.

[12] R. Böhme and G. Kataria, "Models and measures for correlation in cyber-insurance," in *Proc. Workshop Econ. Inf. Secur. (WEIS)*, 2006, pp. 1–26.

[13] H. Herath and T. Herath, "Copula-based actuarial model for pricing cyber-insurance policies," *Insurance Markets Companies: Anal. Actuarial Comput.*, vol. 2, no. 1, pp. 7–20, 2011.

[14] A. Mukhopadhyay, S. Chatterjee, D. Saha, A. Mahanti, and S. K. Sadhukhan, "Cyber-risk decision models: To insure it or not?" *Decision Support Syst.*, vol. 56, pp. 11–26, Dec. 2013.

[15] M. Xu and L. Hua. (2017). *Cybersecurity Insurance: Modeling and Pricing*. [Online]. Available: https://www.soa.org/research-reports/2017/cybersecurity-insurance

[16] M. Xu, L. Hua, and S. Xu, "A vine copula model for predicting the effectiveness of cyber defense early-warning," *Technometrics*, vol. 59, no. 4, pp. 508–520, 2017.

[17] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling multivariate cybersecurity risks," *J. Appl. Stat.*, pp. 1–23, 2018.

[18] M. Eling and N. Loperfido, "Data breaches: Goodness of fit, pricing, and risk measurement," *Insurance, Math. Econ.*, vol. 75, pp. 126–136, Jul. 2017.

[19] K. K. Bagchi and G. Udo, "An analysis of the growth of computer and Internet security breaches," *Commun. Assoc. Inf. Syst.*, vol. 12, no. 1, p. 46, 2003.

[20] E. Condon, A. He, and M. Cukier, "Analysis of computer security incident data using time series models," in *Proc. 19th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Nov. 2008, pp. 77–86.

[21] Z. Zhan, M. Xu, and S. Xu, "A characterization of cybersecurity posture from network telescope data," in *Proc. 6th Int. Conf. Trusted Syst.*, 2014, pp. 105–126. [Online]. Available: http://www.cs.utsa.edu/~shxu/socs/intrust14.pdf

[22] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1775–1789, Nov. 2013.

[23] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1666–1677, Aug. 2015.

[24] Y.-Z. Chen, Z.-G. Huang, S. Xu, and Y.-C. Lai, "Spatiotemporal patterns and predictability of cyberattacks," *PLoS ONE*, vol. 10, no. 5, p. e0124472, 2015.

[25] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling and predicting extreme cyber attack rates via marked point processes," *J. Appl. Stat.*, vol. 44, no. 14, pp. 2534–2563, 2017.

[26] J. Z. Bakdash *et al.* (2017). "Malware in the future? forecasting analyst detection of cyber events." [Online]. Available: https://arxiv.org/abs/1707.03243

[27] Y. Liu *et al.*, "Cloudy with a chance of breach: Forecasting cyber security incidents," in *Proc. 24th USENIX Secur. Symp.*, Washington, DC, USA, 2015, pp. 1009–1024.

[28] R. Sen and S. Borle, "Estimating the contextual risk of data breach: An empirical approach," *J. Manage. Inf. Syst.*, vol. 32, no. 2, pp. 314–341, 2015.

[29] F. Bisogni, H. Asghari, and M. Eeten, "Estimating the size of the iceberg from its tip," in *Proc. Workshop Econ. Inf. Secur. (WEIS)*, La Jolla, CA, USA, 2017.

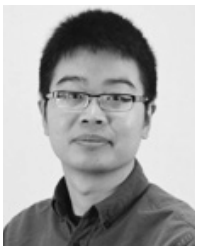
[30] R. F. Engle and J. R. Russell, "Autoregressive conditional duration: A new model for irregularly spaced transaction data," *Econometrica*, vol. 66, no. 5, pp. 1127–1162, 1998.

[31] N. Hautsch, *Econometrics of Financial High-Frequency Data*. Berlin, Germany: Springer-Verlag, 2011.

[32] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling Extremal Events: For Insurance and Finance*. Berlin, Germany: Springer, 1997.

[33] T. Bollerslev, J. Russell, and M. Watson, *Volatility and Time Series Econometrics: Essays in Honor of Robert Engle*. London, U.K.: Oxford Univ. Press, 2010.

[34] R. B. Nelsen, *An Introduction to Copulas*. New York, NY, USA: Springer-Verlag, 2007.
[35] H. Joe, *Dependence Modeling With Copulas*. Boca Raton, FL, USA: CRC Press, 2014.
[36] J. D. Cryer and K.-S. Chan, *Time Series Analysis With Applications in R*. New York, NY, USA: Springer, 2008.
[37] B. Peter and D. Richard, *Introduction to Time Series and Forecasting*. New York, NY, USA: Springer-Verlag, 2002.
[38] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. New York, NY, USA: Springer-Verlag, 2016.
[39] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, vol. 1, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.
[40] M. Y. Zhang, J. R. Russell, and R. S. Tsay, "A nonlinear autoregressive conditional duration model with applications to financial transaction data," *J. Econ.*, vol. 104, no. 1, pp. 179–207, 2001.
[41] L. Bauwens and P. Giot, "The logarithmic ACD model: An application to the bid-ask quote process of three NYSE stocks," *Ann. Économie Stat.*, no. 60, pp. 117–149, Oct./Dec. 2000.
[42] L. Bauwens, P. Giot, J. Grammig, and D. Veredas, "A comparison of financial duration models via density forecasts," *Int. J. Forecasting*, vol. 20, no. 4, pp. 589–609, 2004.
[43] G. W. Corder and D. I. Foreman, *Nonparametric Statistics: A Step-by-Step Approach*. Hoboken, NJ, USA: Wiley, 2014.
[44] P. R. Hansen and A. Lunde, "A forecast comparison of volatility models: Does anything beat a garch(1, 1)?" *J. Appl. Econ.*, vol. 20, no. 7, pp. 873–889, 2005.
[45] S. I. Resnick, *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. New York, NY, USA: Springer-Verlag, 2007.
[46] X. Zhao, C. Scarrott, L. Oxley, and M. Reale, "Extreme value modelling for forecasting market crisis impacts," *Appl. Financial Econ.*, vol. 20, nos. 1–2, pp. 63–72, 2010.
[47] C. Scarrott, "Univariate extreme value mixture modeling," in *Extreme Value Modeling and Risk Analysis: Methods and Applications*, J. Yan and D. K. Dey, Eds. London, U.K.: Chapman & Hall, 2016, pp. 41–67.
[48] H. Joe, *Multivariate Models and Dependence Concepts* (Monographs on Statistics and Applied Probability), vol. 73. London, U.K.: Chapman & Hall, 1997.
[49] H. White, "Maximum likelihood estimation of misspecified models," *Econometrica, J. Econ. Soc.*, vol. 50, no. 1, pp. 1–25, 1982.
[50] W. Huang and A. Prokhorov, "A goodness-of-fit test for copulas," *Econ. Rev.*, vol. 33, no. 7, pp. 751–771, 2014.
[51] W. Wang and M. T. Wells, "Model selection and semiparametric inference for bivariate failure-time data," *J. Amer. Statist. Assoc.*, vol. 95, no. 449, pp. 62–72, 2000.
[52] C. Genest, J.-F. Quessy, and B. Rémillard, "Goodness-of-fit procedures for copula models based on the probability integral transformation," *Scandin. J. Stat.*, vol. 33, no. 2, pp. 337–366, 2006.
[53] A. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[54] P. F. Christoffersen, "Evaluating interval forecasts," *Int. Econ. Rev.*, vol. 39, no. 4, pp. 841–862, 1998.
[55] R. F. Engle and S. Manganelli, "CAViaR: Conditional autoregressive value at risk by regression quantiles," *J. Bus. Econ. Stat.*, vol. 22, no. 4, pp. 367–381, 2004.
[56] P. M. Romer, "Increasing returns and long-run growth," *J. Political Econ.*, vol. 94, no. 5, pp. 1002–1037, 1986.
[57] G. M. Ljung and G. E. P. Box, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978.
[58] G. R. Shorack and J. A. Wellner, *Empirical Processes With Applications to Statistics*. Philadelphia, PA, USA: SIAM, 1986.
[59] M. A. Stephens, "Tests based on EDF statistics," in *Goodness-of-Fit Techniques*, R. B. d'Agostino and M. A. Stephens, Eds. New York, NY, USA: Marcel Dekker, 1986, pp. 97–193.



Maochao Xu received the Ph.D. degree in statistics from Portland State University in 2010. He is currently an Associate Professor of mathematics with Illinois State University. His research interests include statistical modeling, cyber risk analysis, and ensuring cyber security. He also serves as an Associate Editor for *Communications in Statistics*.

Kristin M. Schweitzer is a Mechanical Engineer with the U.S. Army Research Laboratory (ARL), Cyber and Networked Systems Branch. Her current role is to conduct and coordinate use-inspired basic research in cyber security for the ARL South office located at the University of Texas at San Antonio. Previously for ARL, she provided Human Systems Integration analyses for U.S. Army, Marine Corps, Air Force, and Department of Homeland Security systems. She also conducted research on human performance in uncontrolled environments.

Raymond M. Bateman received the Ph.D. degree in mathematical and computer sciences (operations research) from the Colorado School of Mines. He retired as a Lieutenant Colonel from the U.S. Army Special Forces with 20 years of enlisted and officer service. He conducted research for significant and relevant issues affecting the U.S. Army Medical Department Center and School, Health Readiness Center of Excellence by applying human systems integration (HSI) and operations research techniques. He currently serves as the Army Research Laboratory (ARL) South Lead for cybersecurity for use-inspired basic research at The University of Texas, San Antonio. His projects included serving as the Non-Medical Operations Research Systems Analyst and HSI Expert for the Medical Command Root-Cause Analysis Event Support and the Engagement Team that investigates sentinel events that result in permanent harm or death. He has two deployments to Iraq as the Army Civilian Science Advisor to Commander III Corps and Army Materiel Command.

Shouhuai Xu received the Ph.D. degree in computer science from Fudan University. He is currently a Full Professor with the Department of Computer Science, The University of Texas at San Antonio. He is also the Founding Director of the Laboratory for Cybersecurity Dynamics. He pioneered the Cybersecurity Dynamics framework for modeling and analyzing cybersecurity from a holistic perspective. He is interested in both theoretical modeling and analysis of cybersecurity and devising practical cyber defense solutions. He co-initiated the International Conference on Science of Cyber Security (SciSec) in 2018 and the ACM Scalable Trusted Computing Workshop. He is/was a Program Committee Co-Chair of SciSec'18, ICICS'18, NSS'15, and Inscrypt'13. He was/is an Associate Editor of IEEE TDSC, IEEE T-IFS, and IEEE TNSE.