

## An Intelligent System for Cybercrime Detection and Control

David, D.C., Anireh, V.I.E & Matthias, D.

Department of Computer Science  
Rivers State University  
Port Harcourt, Nigeria

**E-mails:** chidondave@gmail.com, anireh.ike@ust.edu.ng, pastordanielmatthias@yahoo.com

**Phones:** +234-8172070291, +234-8033229172, +234-8132249596

### ABSTRACT

This paper presents an intelligent system for cybercrime detection and control. A genetic approach was used for the detection and removal of cross-site scripting vulnerabilities in the source code of PHP and JavaScript. Malicious code detection filters JavaScript and PHP codes, text enclosed with tags are seen as code, any untrusted data referenced as a quoted data value in a JavaScript block and PHP block are trapped and filter by the system, these codes cannot be executed as they were detected and removed by genetic approach technique hence ascertains the efficiency of the system. Result shows that all tried User IP was blocked after three attempts using lock account technique and the system detects a malicious code that could gather cookie data and redirect the cookie data to another malicious website; the JavaScript code was detected and removed by the system and could not be executed. The system was implemented in Python programming language.

**Keywords** – Intelligent System, System Detection, System Control, Lock Account, Genetic Approach

---

### CISDI Journal Reference Format

David, D.C., Anireh, V.I.E & Matthias, D. (2019): An Intelligent System for Cybercrime Detection and Control. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 10 No 1.Pp 35-46 Available online at www.cisdijournal.org  
DOI Affix - <https://doi.org/10.22624/AIMS/CISDI/V10N2P3>

---

### INTRODUCTION

Cybercrimes are crimes committed over the internet. Cybercrimes are offences or acts performed against persons or an organization with a wrong motive to advisedly hurt the person or the organization exploitation trendy telecommunication networks, web like emails, websites (Halder et al., 2012). However, all cybercrimes comprise the system (computer) and the person using it as victims; the main target depends totally on which of the two factors. Hence, the system is considered to be either a tool for the sake of easiness. For example, the attacking of the computer's information and distinct resources is done by hackers. The vital thing to note is that overlapping happens in several instances. This study develops an intelligent system for cybercrime detection and control model using lock account and genetic approach techniques to block attackers from gaining access into the system, conjointly detect and remove malicious code.

#### 1.1 Statement of Problem

Cyber security is quite challenging due to varied degrees of security features within the cloud entities in the cyberspace. Majority of cybercriminal activities do not involve physical damage or stealing of equipment, but are rather intellectual manipulations which are very difficult to detect and control (Jordan & Taylor, 2004). Some of the problems and challenges of cybercrime are as follows: Hactivism, Ransom Ware, malware links and files and the threats from an insider.

## 1.2 Aim and Objective

The aim of this research is to design an intelligent system for cybercrime detection and control and the objectives are to Identify data used for cyber-attack using genetic algorithm, design a model to detect attacker using Lock Account and Genetic Approach techniques, and develop a system to identify unauthorized access using Python.

## 2. RELATED WORKS

Hight (2015) designed Cybercrime Detection and Control Using the Cyber User Identification Model to recognize cyber user and control cybercrime. The approach used is the object-oriented example of system analysis and design. The crime situations thought-about for finding are phishing, fraud and information theft. The process for the usage of the model is PHP and java. My Structured Query language (MySQL) was utilized as the database. The hardware utilized for usage has inbuilt webcam or connected advanced camera for facial picture catching, a Global Positioning System (GPS) sensor to detect a cyber-user's location, and a unique mark scanner. The work is demonstrated to supply interfaces to catch the digital signatures for every data sent to the internet, the client's fingerprints and facial picture as login parameters, determine and record the position of the cyber client, the Media Access control (MAC) address of the system utilized, the date, time and also the reasonably activity meted out by the cyber client while on the web, at that point record security dangers for an analysis by cybercrime agents.

The results showed that the system will really recognize the cyber user and his/her crimes while on the web. Wijesinghe et al (2016) proposed a Combating Cyber Crime Using Artificial Agent Systems by Merging Genetic Algorithm and Fuzzy Logic to detect interruptions using an Agent Communication Language (AGL) for purpose of communication. In making new methodologies, Genetic Algorithm and Fuzzy Logic Algorithm are being utilized. Genetic Algorithm is a streamlining calculation that helps in finding proper fuzzy principles. This study showed that there is a vitality of a high security way to deal with safe information and assured communication of data between various establishments. Maria (2015) developed an Understanding and Defending Against Internet Infrastructures Supporting Cybercrime Operations, the type of infrastructure presented is fast-flux service network which is based on Domain Name System (DNS) manipulation. The methodology used is empirical studies that help to advance understanding, about how these infrastructures operate.

The study showed that the system can help to counteract cybercriminal infrastructures. Shah et al., (2017) proposed an Intrusion Detection System-Types and Prevention: This research was developed using anomaly-based detection to see organize traffic and recognizes information that's erroneous, invalid, or usually irregular. This methodology is helpful for detection of undesirable traffic that is not explicitly known. For example, irregularity-based IDS can recognize that an Internet Protocol (IP) packet is distorted. It doesn't identify that it is distorted during a particular method, however shows that it is abnormal. Sung et al (2014) developed a Big Data Analysis System Concept for Detecting Unknow Attacks: Another model dependent on huge information examination procedures that can extricate data from an assortment of sources to recognize future attack. System based on future Advanced Persistent Threat (APT) identification and avoidance system implementation to shield against these obscure attacks, which can't be identified with existing innovation or expertise.

### 3. METHODOLOGY

#### 3.1

#### 3.2 Proposed System

The proposed (Intelligent System for Cybercrime Detection and Control) system is based on Brute Force Attack and Cross-Site Scripting using lock account and genetic approach techniques.

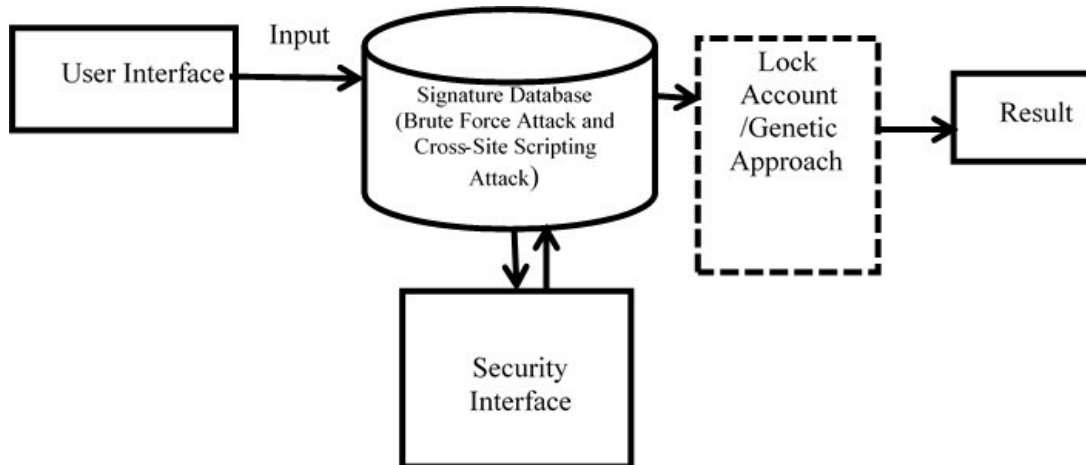


Figure 1: Proposed System Architecture

The crime scenario considered for the proposed system are Brute force attack and Cross-site Scripting. The input variables of BFA and XSS get into the Cybercrime Detection and Control System, the rules in the system counteract on the input variables using lock account and genetic approach to produce results (that is, blocking unauthorized attempts and disable malicious scripts).

#### 3.3 Description of the components

This section gives the description of the components used for the system

**User Interface:** This is the crossing points of system users' such as administrator, authorized users and unauthorized users (hackers) into the system. The user gains access to the system through submitting his login details by using this form.

**Signature Database:** contains observed data with predefined descriptions of intrusive behavior. A database of previous attack signatures and known system vulnerabilities. It recognizes patterns typical of attacks and analyzing system configurations and vulnerabilities.

**Security Interface:** Is used by system administrator to set block to Internet Protocol (IP) addresses from unauthorized users several attempts.

**Lock Account:** The user lock threshold specifies the number of invalid logins attempts that are allowed before user account is locked or blocked.

**Genetic Approach:** Is a method used to identify the key components or characteristics responsible for intrusion.

**3.4 Details of Experiment**

Cyber-crime detection and control system was tested in Window Operating System, Linux and Android Operating System, the system was developed using Python programming language. The system was built to detect “Trial and Error” by guessing usernames and passwords, and malicious code such as JavaScript codes and PHP codes used by hackers. The system requires user authentication to go through it, in the event that the hacker attacks based on accurate lexicon words, at that point system detects the attempts, then if the hacker slightly changes the lexicon words; the system detects it. Thus, the system detects both dictionary and Hybrid Brute-force attacks. Parameters used here are username and password, text and codes (JavaScript and PHP)

**Table 1: Input parameters**

Input Parameters	Description
Username	User identification name
Password	User secret pin
Text	Words or sentence in English language
JavaScript code	Codes written in JavaScript to get username/password
PHP code	Codes written in PHP language to get username/password

These input parameters allow access into the system.

**3.5 Using Lock Account against BFA**

Developing a rule that enables setup of solely brute-force resistant password. Length, Cardinality, and Entropy are three parameters that are adopted for password resilience against BFA. This least password strength prerequisite is twelve characters. These twelve characters passwords measure of strength in bits. Password entropy may be an unpredictable measurement. The entropy formula is given below:

$$E = \log_2 R^L \dots\dots\dots (1)$$

Where,

E = password entropy

L = number of characters in your password

R = pool of unique characters, (that is combination of lower case, upper case, digit and special character).

$R^L$  = cardinality; the number of possible passwords

$\log_2 R^L$  = the number of bits of entropy

#### 4. RESULT

The combination of cardinality increases the strength of password. Combination of digit and lower-case alphabet gives a cardinality of 36. Combination of digit, lower case alphabet and upper-case alphabet gives a cardinality of 62. Combination of digit, lower case alphabet, case alphabet and special characters gives a cardinality of 94. Table 2 shows the password strength. Password length of 12 and cardinality (10, 26, 32, 36, 62).

**Table 2: Password Strength**

S/N	LENGTH	CARDINALITY	ENTROPY (PASSWORD STRENGTH) Bits
1	6	94	39.3
2	8	94	52.4
3	12	10	39.9
4	14	36	72.4
5	12	26	56.4
6	12	26	56.4
7	12	32	60.0
8	12	36	62.0
9	12	62	71.5
10	12	94	78.7
11	14	94	91.8
12	32	94	209.7

Combination of length (12) and cardinality (10) gives entropy of 39.9 bits which gives weak password, while password length of 12 and cardinality of 94 gives entropy of 78.9 bits which gives a strong password. Account information of authorized and non-authorized user in Window and Android Operating System as shown in Table3.

**Table 3: Hackers Attempt Log Page (Window And Android OS)**

S/N	ATTEMPT TIME	IP ADDRESS	USER AGENT	USERNAME	PATH	LOGIN VALID
1	March 20, 2019, 7:42 p.m	10.240.1.220	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.121 Safari/537.36	project	/admin/login/	True
2	<u>March 9, 2019, 4:47 p.m.</u>	10.240.0.239	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36	gift	/accounts/login/	False
3	<u>March 9, 2019, 4:47 p.m.</u>	10.240.0.227	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36	gift	/accounts/login/	False
4	<u>March 9, 2019, 3:52 p.m.</u>	10.240.0.227	Mozilla/5.0 (Linux; Android 8.0.0; Infinix X573) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.105 Mobile Safari/537.36	ugochukwu	/accounts/login/	True
5	<u>Feb. 11, 2019, 7:41 p.m.</u>	10.240.0.87	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36	web	/accounts/login/	False
6	<u>Feb. 11, 2019, 7:41 p.m.</u>	10.240.0.185	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36	web	/accounts/login/	False
7	<u>Feb. 11, 2019, 7:41 p.m.</u>	10.240.1.220	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36	web	/accounts/login/	False
8	<u>Feb. 11, 2019, 7:40 p.m.</u>	10.240.1.97	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36	web	/accounts/login/	False
9	<u>Feb. 7, 2019, 10:42 a.m.</u>	10.240.0.87	Mozilla/5.0 (Linux; Android 4.2.2; TECNO S9 Build/JDQ39) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36	info	/accounts/login/	False
10	<u>Feb. 7, 2019, 10:42 a.m.</u>	10.240.0.185	Mozilla/5.0 (Linux; Android 4.2.2; TECNO S9 Build/JDQ39) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36	info	/accounts/login/	False
11	<u>Feb. 7, 2019, 10:42 a.m.</u>	10.240.0.239	Mozilla/5.0 (Linux; Android 4.2.2; TECNO S9 Build/JDQ39) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36	info	/accounts/login/	False
12	<u>Feb. 7, 2019, 10:41 a.m.</u>	10.240.0.187	Mozilla/5.0 (Linux; Android 4.2.2; TECNO S9 Build/JDQ39) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36	info	/accounts/login/	False
13	<u>Jan. 26, 2019, 2:07 p.m.</u>	127.0.0.1	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36	ugochukwu	/accounts/login/	True

The System was tested in Windows, Android and Linux Operating system. Table 3 shows the hacker's login attempt, location, time and date, IP Address, user agent, username, path and login valid in Windows and Android OS. This table captures the details of the malicious hacker, the operating system used to execute the hack, the type of browser and its version, the time and date of the hack attempt and as well as the location of the hacker which can be further viewed in a map which points to the exact location of the hacker. Result shows that there were 13 attempts and 3 true logins from "Hackers Attempt Log Page (Window and Android OS) table" and 10 false attempts via the account login path.

**4.1 Results using Genetic Approach against XSS**

Cross-site scripting works by injecting code into the injection point in web application, such as the search field, feedback forms. The system accepts text document and JavaScript and PHP codes, Text enclosed with tags are seen as code, any untrusted data referenced as a quoted data value in a JavaScript block and PHP block are trapped by the system.

**Table 4. Malicious Code Detection**

S/N	CODE FILTER
1	<script>document.location=malicious_http//evil.com?cookie="+document.cookie</script>
2	<script>document.getElementById("password")</script>
3	<script>document.getElementById (password)< script>
4	getElementById()

Code filter for cross-site scripting in Table 4. shows that the system detected JavaScript code to get element password, JavaScript code redirecting to malicious URL and code to get element by ID. The first row (1) has a JavaScript code that could be used to gather cookie data from the website and redirect the cookie to malicious website "http//evil.com?cookie" for example, session IDs or Login information. Second row (2) has a JavaScript code to get element by password "document.getElementById("password)". These codes were not executed as they were detected and removed by genetic approach stated. Malicious code detection filters text document, JavaScript and PHP codes. Text enclosed with tags are seen as code, any untrusted data referenced as a quoted data value in a JavaScript block and PHP block are trapped and filter by the system.

The graphical representation of hacker login attempt in Table 3 is shown in Figure 2. The result of this graph is shown below.

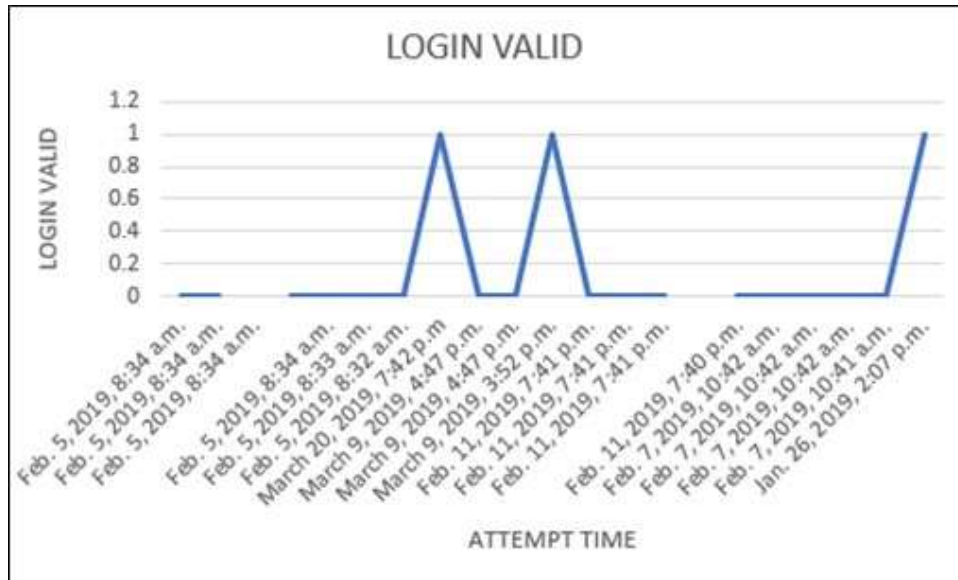


Figure 2: Login Valid

Figure 2 represents the login details of authorized and unauthorized users. A graph of "Login Valid" against "Attempt Date". "Attempt Time" is on x-axis and "Login Valid" is on y-axis, this shows the False login as zeros (0) and True login as ones (1), there were 3 True logins having one value (1) (March 20, 2019, 7:42 p.m., March 9, 2019, 3:52 p.m., Jan. 26, 2019, 2:07 p.m.) while others are false login.

**5 CONCLUSIONS**

In this study, the Cybercrime Detection and Control system was developed using Lock Account and Genetic Approach. Lock Account technique was used to detect unauthorized attempts to the system and attackers (hackers) IP address was blocked, hence, denying hackers access into the system. Genetic Approach was used to detect and remove malicious code that could grant hackers access into the system. This technique filters JavaScript and PHP codes and stops code from executing. Such codes were seen as text by the system.



## REFERENCE

1. Abubakar, A. & Pranggono, B. (2017). *Machine learning based intrusion detection system for software defined networks*. In: Proceedings of the 2017 Eighth International Conference on Emerging Security Technologies (EST): 4(13) 1-7.
2. Chaudhari, R. R. & Patil, S. P. (2017). A study on data mining and machine learning for intrusion detection system. *International Journal of Advanced Research in Computer/Communication Eng* 6(2), 114-118.
3. Claudio Comis Da Ronco, & Ernesto Benini (2013). A Simplex-Crossover-Based Multi-Objective Evolutionary Algorithm, IAENG Transactions on Engineering Technologies, Volume 247 of the series Lecture Notes in Electrical Engineering 7(4) 583-598
4. Crikovic, G. D. (2010). *Constructive research and info-computational knowledge Generation*, Springer, Berlin, Heidelberg, 314.
5. Denning, D. (2001). Activism, Hactivism, and Cyber terrorism: The Internet as a tool or Influencing Foreign Policy. In Arquilla, J. and Ronfeldt, D. (ed.), *Networks and Netwars*. Rand, USA, p. 241.
6. Hight, C. I. & Moses, A. A. (2015). Cyber crime detection and control using the cyber user identification model. *IRACST - International Journal of Computer Science and Information Technology & Security (IJSITS)*, 5(5), ISSN: 2249-9555.
7. Kroenke, L. W. (2008). Pacific absolute plate motion since 145 Ma: An assessment of the fixed hot spot hypothesis. *Journal of Geophysical Research: Solid Earth*, 113(B6).
8. Kumar, B. S., Raju, T. C. S. P., Ratnakar, M., Baba, S. K. D. & Sudhakar, N. (2013). Intrusion Detection System-Types and Prevention. *International Journal of Computer Science and Information Technologies*, 4 (1), 77– 82.
9. Lotfi, A. Z. (1994). The role of fuzzy logic in modeling, identification and control. University of California at Berkeley, Retrieved from <http://www.mic-journal.no/abs/mic-3-9.asp>. 15(3), 191-203.
10. Maguire, M. (2000), Policing by risks and targets: Some dimensions and implications of intelligence-led crime control", *Policing and Society*, pp.3.
11. Maria, K. (2015). *Understanding and defending against internet infrastructures supporting cybercrime operations*. USA: Georgia Institute of Technology.
12. Mariam, N., Jason, R. C. N., & Michael, G. (2016). *Towards Designing a Multipurpose Cybercrime Intelligence Framework*. Department of Computer Science, University of Oxford, UK.
13. Moschytz, G., & Dumitras A. (2007). Understanding fuzzy logic: An interview with Lotfi Zadeh [DSP History]. *IEEE Signal Processing Magazine*, 24(3), 102 – 105.
14. Nouh, M., Nurse, J. R., & Goldsmith, M. (2016). Towards Designing a Multipurpose Cybercrime Intelligence Framework. Paper presented at the Intelligence and Security Informatics Conference (EISIC).
15. Perez, D. Astor, M. A. Abreu, D. P & Scalise, E. (2017). Intrusion Detection in Computer Networks Using Hybrid Machine Learning Techniques. Central University of Venezuela, Caracas, Venezuela, pp1-10
16. Rajarshi, R. C., Somnath B., & Digbijay G., (2013). Cyber crimes- challenges & solutions. *International Journal of Computer Science and Information Technologies(IJSIT)*, 4(5), 729-732.
17. Shah, R. A., Qian, Y., Kumar, D., Ali, M. & Alvi, M. B. (2017). Network Intrusion Detection through Discriminative Feature Selection by Using Sparse Logistic Regression. *Future Internet*, 9(81), 1-15.
18. Sung, H. A., Nam, U. K., & Tain M. C. (2014). Big Data Analysis System Concept for Detecting Unknown Attacks. College of Information and Communication Engineering, Sungkyunkwan University, South Korea.
19. Vijayarani, S. & Sylvia, S. M. (2015). Intrusion Detection System – a study. *International Journal of Security, Privacy and Trust Management (IJSPTM)*, 4(1), 31-44.
20. Wijesinghe, L.S., De Silva, L.N.B., Abhayaratne, G.T.A., Krithika P., Priyashan S.M.D.R., & Dhammearatchi D. (2016). Combating Cyber Crime Using Artificial Agent Systems. Sri Lanka Institute of Information Technology Computing. *International Journal of Scientific and Research Publications*, 6(4), 2250-3153.

**APPENDIX A (TRAINING DATA)**

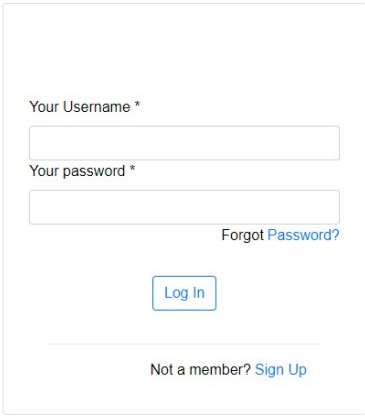
**TABLE A 1 For Lock Account Technique**

<b>Username</b>	<b>Password</b>
Info	Info
Info	Info123
Info	123456
Web	12345
Test	Test1234
Qwerty	Qwerty
Admin	Admin123
Admin	Admin
Root	root

**TABLE A 2: For Genetic Approach**

<b>Codes</b>
<script>document.location=malicious_http//evil.com?cookie="+document.cookie</script>
<script>document. Location=malicious URL</script>
<script>document. getElementById(password) </script>
<script>document. getElementById" password" </script>

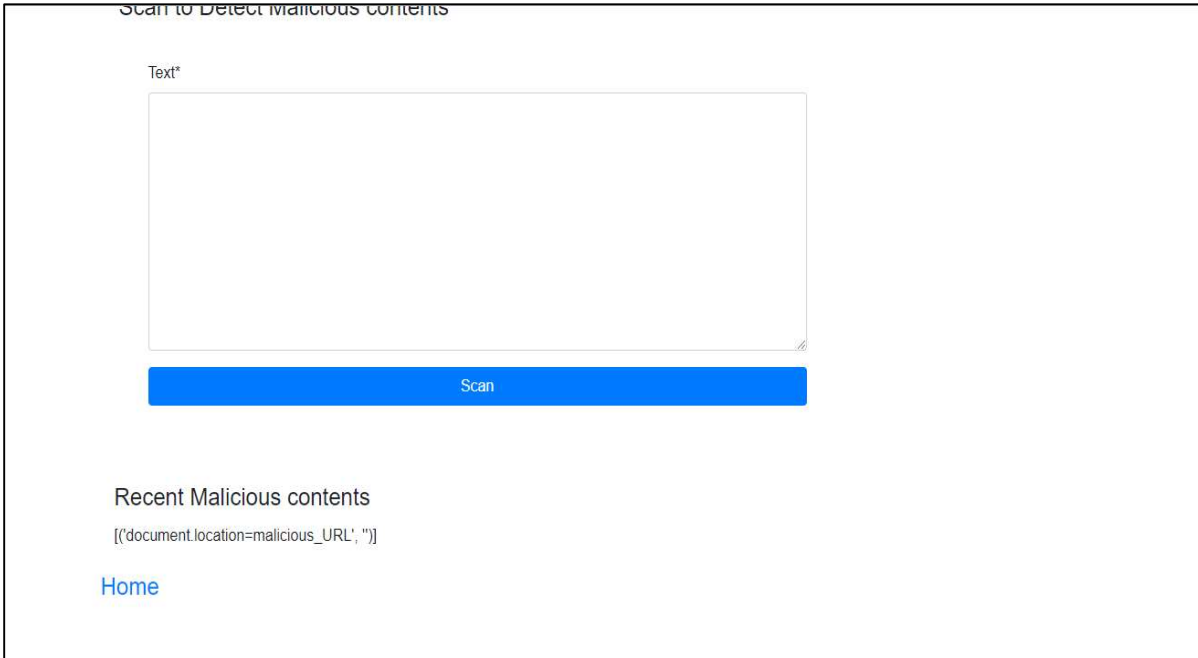
APPENDIX B



A user login form with the following elements:

- Label: "Your Username \*"
- Input field for username
- Label: "Your password \*"
- Input field for password
- Link: "Forgot Password?"
- Button: "Log In"
- Text: "Not a member? [Sign Up](#)"

Figure B 1: User login page



A form titled "Scan to Detect Malicious contents" with the following elements:

- Label: "Text\*"
- Large text area for input
- Blue button: "Scan"
- Section: "Recent Malicious contents"
- Code snippet: `[('document.location=malicious_URL', '')]`
- Link: "Home"

Figure B 2: Text Area

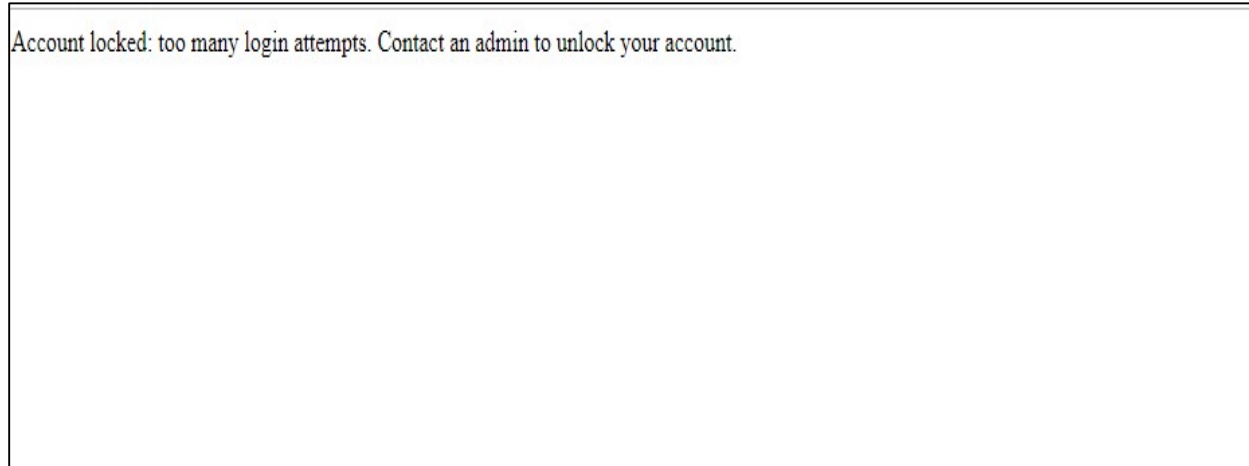


Figure B 3: Account Locked

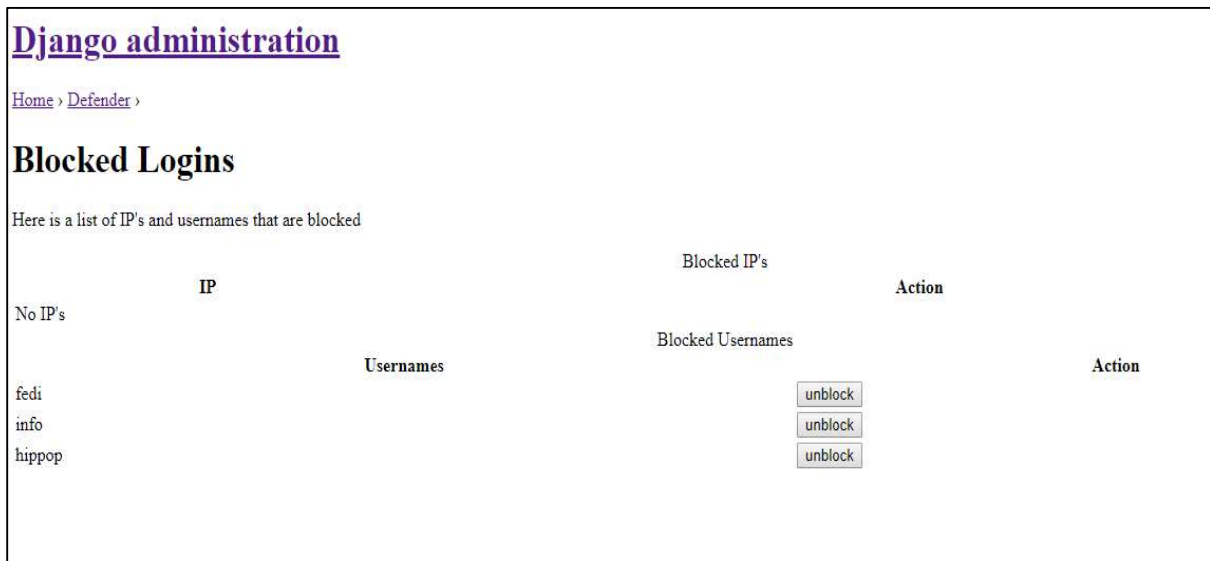


Figure B 4: Blocked Users