Honeypot Deception Method for Real Time Intrusion Detection and Prevention

A Disaster Recovery and Business Continuity Management Report

BCI-3002

submitted by

Project Members

Khushi Sharma	**20BCI0243**
Kaif Ahmad	**20BCI0217**
Nabil Ashraf	**20BCI0279**

Submitted to

Prof. Somasundaram S K

in partial fulfilment for the award of the degree of

Bachelor Of Technology

in

Computer Science

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

MARCH 2023

Table of contents

1. Abstract

Honeytrap is a security instrument intended to identify and forestall assaults on TCP and UDP administrations. It works as a foundation daemon and dispatches server processes when an association endeavor is made to a port. The instrument utilizes different strategies for activity to deal with associations, including sending conflicting information to impersonate notable conventions like TCP and UDP. This befuddles many robotized assault devices, which then, at that point, go on with the assault. To upgrade the instrument's usefulness, a module has been fostered that empowers custom expansions to be handily composed and stacked into the device. The device gathers assault information into an assault string, which can be saved to documents or a SQL data set for manual examination and further activity.

Keywords: Honeypot, IDS, TCP and UDP protocols, SQL database

2. Introduction

The internet has been susceptible to a range of attacks and intrusion attempts for a significant period. These attacks can take various forms, including weak passwords and SQL injections. It is alarming to note that in 2021, there are approximately 929 million open devices, a significant portion of which have either empty or default credentials, making them vulnerable to exploitation.

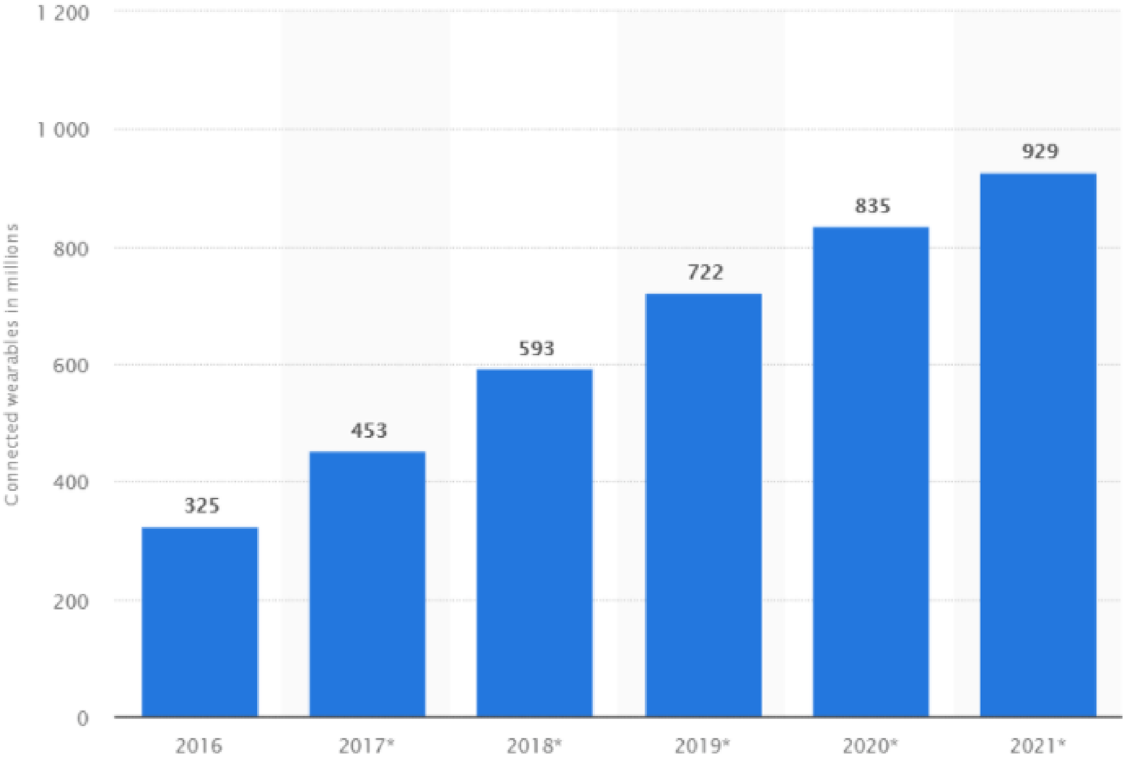


Figure 1 - Number of connected devices worldwide 2016-2021. (in millions)

The quantity of web clients is developing quickly, and with this increment comes an increased gamble of pernicious interruptions. To battle this, different frameworks are being presented, including honeypots, which are traps set in PC innovation to recognize unapproved admittance to delicate information. The objective of a honeypot is to accumulate however much data as could be expected about a gatecrasher's example, programs utilized, and reason for assault in an undercover way. By recognizing gatecrashers, a honeypot creates a log that reports any nosy action and assists with diminishing information break dangers.

This task centers around a comparative design, with the additional component of taking screen captures of the client's framework as well as recording their IP address and area. The honeypot can be introduced inside or outside the organization, or in the DMZ. Any move made by a gatecrasher inside the honeypot can be recorded, including access endeavors, caught keystrokes, and recognizable proof of documents and projects that have been gotten to and refreshed. This data can be utilized to decide an aggressor's definitive thought processes on the off chance that they are uninformed they are inside a honeypot.

A honeypot is a profoundly gotten processing asset that is intended to be tested, interfered with, went after, or split the difference. Honeypots are utilized for two purposes: early admonition and criminological investigation. They are not difficult to set up and are more compelling than different frameworks at catching programmers and malware. With just a solitary connect to it, a honeypot can identify and perceive the interloper. Moreover, a honeypot can catch and segregate malware and assailant assets, then report back to the client in a couple of days with an arrangement in light of the information gathered by the aggressors.

3. Literature survey

Sr. No.	Paper Title and Author (APA)	Methodology
1	Provos, N. (2004, August). A Virtual Honeypot Framework. In USENIX Security Symposium (Vol. 173, No. 2004, pp. 1-14).	The paper presents a light end framework to make a Honeypot, called Honeyd. The approach they use includes receiving and replying to network packets. Its architecture consists of a database, packet dispatcher, protocol handlers, personality engine and a routing component. Honeyd can simulate the networking stack of different OS and can provide different routing topologies and services for a virtual system. They conclude that Honeyd framework can be used in many different areas of System security
2	Krawetz, N. (2004). Anti-honeypot technology. IEEE Security & Privacy, 2(1), 76-79.	The paper discusses various tools and technologies used against HoneyPot. They discuss how honeypot technology is detectable and minimal servers provide an open port. Paper then proceeds to discuss tools like HoneyPot Hunter tests open proxy connectivity and how it helps detecting Honeypot thus helping malicious users to bypass a HoneyPot. It then discusses technology to detect a HoneyPot Hunter using network connection methods and improvements to make it further undetectable.
3	Naik, N., Jenkins, P., Savage, N., & Yang, L. (2021). A computational intelligence enabled honeypot for chasing ghosts in the wires. Complex & Intelligent Systems, 7(1), 477-494.	The paper discusses methods to chase ghosts in wires using HoneyPot. The methodology focuses on identifying abnormalities/patterns in the various fields of TCP/IP protocols as signs of OS fingerprinting attacks, Discovering abnormalities/patterns in TCP Flags ,URG/PSH/FIN probing, NULL packet probing, Reserved Bit Probing, ECN- echo probing, FIN Probing, Discovering abnormalities/patterns in TCP Options, Discovering abnormal/frequent uses of TCP urgent pointer, Discovering abnormalities/variations in TCP Window Size, Discovering abnormalities/commonalities in IP identification (IPID) field etc. As the attacker's traffic enters its analyzed by HoneyPot,analyzed for indicative fields as a sign of an Os Fingerprintingattack and Prediction is presented as result
4	Nazario, J.(2009). PhoneyC: A Virtual Client Honeypot. LEET, 9, 911-919.	In the past few years many client side attacks have been increasing so honeypots enables deep research into server-side attacks, honey-clients can permit the deep study of client-side attacks. In this paper they have told about PhoneyC, they have suggested a honey client tool which provides a visibility into new and complex client-side attacks. PhoneyC is a virtual honey client. Their tool can remove the obfuscation from many

		malicious code or pages. It also emulates specific vulnerabilities to pinpoint the attack vector. It is a modular framework which has enabled to study of malicious HTTP code or pages and from that we can understands advance vulnerabilities and attacker techniques
5	Seifert, C., Welch, I., & Komisarczuk, P. (2007). Honeyc-the low-interaction client honeypot. Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 6.	The author in this paper has discussed that honeypots is a security device which can detect the malicious code itself. By capturing the such code we can able to understand the operations and activities of the attacker or hacker. In this paper they have discussed about how Server Honeypot Architectures works and also about the Client Honeypot Architectures. They have introduced a new type of client honeypot which is HoneyC, which is low interaction client honeypot which is designed to address some of the shortcomings of traditional high interaction client honeypots. They have acknowledged that it is signature-based detection which was implemented in HoneyC which is likely to give false alerts.
6	Mairh, A., Barik, D., Verma, K., & Jena, D. (2011, February). Honeypot in network security: a survey. In Proceedings of the 2011 international conference on communication, computing & security (pp. 600-605).	They have discussed about they honeypot, as we can see due to rapid growth in internet, people can gain access of important information about anyone messages and can use it for bad purpose by the attacker so they have reviewed about the recent advancement in the honeypot. They have gone through some notable idea regarding honeypots.They have also discussed about the analysis of the of honeypot. they have explained about the aspects of using honeypot in education and in hybrid environment with IDS. They have defined about the signature used in technique of analysis of traffic in honeypot
7	Zhang, F., Zhou, S., Qin, Z., & Liu, J. (2003, August). Honeypot: a supplemented active defense system for network security. In Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies (pp. 231-235). IEEE.	In this paper, they have reviewed the technical process and security contribution of honeypot production and research about different honeypot. They have also suggested typical honeypot solutions. They have suggested to us how to implement the technical trends of how to configure, integrate, virtualize, and distribute working honeypot for the future. As we know honeypot is an active defense system in a security network. It can trap different attacks, record all intrusion information, what all tools are used, and the activities of the hacking process, and it ensures it prevents attacks. Out of many security solutions, a honeypot gives solutions to many traditional attacks or dilemmas.

8	Alata, E., Nicomette, V., Kaâniche, M., Dacier, M., & Herrb, M. (2006, October). Lessons learned from the deployment of a high-interaction honeypot. In 2006 Sixth European Dependable Computing Conference (pp.39-46). IEEE.	In this paper they have carried out an experimental study and they found out by observing the attack of the hackers when login in a compromised machine. They had done this for a six months period and concluded that during which a controlled experiment was going on and running with a high interaction honeypot. they have correlated their findings with those obtained with a worldwide distributed system of low-interaction honeypots which were existing.
9	Kuwatly, I., Sraj, M., Al Masri, Z., & Artail, H. (2004, July). A dynamic honeypot design for intrusion detection. In The IEEE/ACS International Conference onPervasive Services, 2004. ICPS 2004. Proceedings. (pp. 95-104). IEEE.	Honeypot gives an idea and platform to study the different methods and tools used by attackers, especially the black hat hacker community. Thus, we can know about their value from the unauthorized use of the resources. This modern technology specially in the area of Intrusion Detection is a honeypot that tends to provide all the things which are required for successful attack completion. In this paper they have discussed about design of a dynamic honeypot, which is an autonomous honeypot capable of adapting in a dynamic and constantly changing network environment. This approach has given integration to active or passive probing and virtual honeypots. They have addressed about the challenges of configuring it and configuring Virtual honeypots.
10	Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., & Rossow, C. (2016). IoTPOT: A novel honeypot for revealing current IoT threats. Journal of Information Processing, 24(3), 522-533.	In this paper they have analyzed threats present in IOT devices. They have shown Telnet based attacks which are targeting the IOT devices. Using this have proposed an IOT honeypot and sandbox which attracts and analyzes the telnet based attacks on various different IOT devices on different architectures like ARM, MIPs. Also they have given results of their observation of their honeypot and capture the malware samples. They have tried 5 different DDoS malware families targeting Telnet-enabled IOT devices and have also tried on 9 different CPU architectures.

4. Existing System

The earliest realized honeypots were gadgets with virtual machines and various IP addresses. In a 2004 paper, it was uncovered that these honeypots were primarily carried out at the organization level and had restricted use in catching association and compromise endeavors. One significant downside of these frameworks was that clients were always unable to acquire full admittance to the framework programming [1].

In the years 2006-2007, the center moved towards making honeypots more intuitive and further developing the client experience. A 2007 paper exhibited how before honeypots were fundamentally centered around going after instead of following way of behaving to decide if the client was a noxious one or an innocuous one who committed errors. The Honeypot utilized here, called HoneyC, utilized an investigation motor to follow client conduct and assault appropriately [7].

By 2009, server-side assaults were on the ascent, and a progression of facilitated assaults in July of that year shook significant organizations and constrained them to rethink their security frameworks. Honeypots turned into a pivotal need, developing to cover increasingly more server-side assaults. In a 2009 paper, profound examination into server-side assaults, honey-clients had the option to permit the profound investigation of client-side assaults. The paper presented PhoneyC, a virtual honeyclient device that gives perceivability into new and complex client-side assaults. PhoneyC is a measured structure that empowers the investigation of noxious HTTP code or pages, considering a superior comprehension of cutting edge weaknesses and assailant procedures [4].

From 2011 to 2016, there was huge investigation into honeypots, with digital activists talking about ways of assisting honeypot innovation. Because of the fast development of the web, individuals can undoubtedly get to delicate data and use it for odious purposes, making the survey of ongoing honeypot progressions important. Outstanding thoughts were introduced in regards to honeypots, including examination of honeypot viewpoints, honeypots in schooling, and their utilization in crossover conditions with IDS. They additionally characterized the mark utilized in breaking down traffic in honeypot Telnet-based assaults that target IoT gadgets, proposing an IoT honeypot and sandbox that draws in and dissects telnet-put together goes after with respect to different IoT gadgets on various structures. They likewise introduced the aftereffects of their honeypot perceptions and malware tests, testing five unique DDoS malware families focusing on Telnet-empowered IoT gadgets on nine different computer processor designs.

Current honeypots are lightweight, intuitive codes running on frameworks that evaluate and examine client conduct and assault appropriately. They are profoundly intelligent, with appropriate web applications to track and send off computerized assaults. At the point when an aggressor's traffic enters, it is investigated by the HoneyPot, and examined for characteristic fields as an indication of an operating system fingerprinting assault, and the expectation is introduced subsequently [3].

5. Software and Hardware Requirements:

SOFTWARE:

Our HoneyPot, which we have named HoneyTrap, is an exceptionally powerful device for catching aggressors. It is a lightweight Python code that uses different Python modules like Cup, cv2, Numpy, ImageGrab, PIL, Jsonify, Flask_mail, Datetime, and Hashlib. The code has a direct way of execution. In the first place, it acknowledges a hashed secret phrase as a seed esteem. At the point when a client enters their certifications, the code checks in the event that the qualifications are right. Assuming that the qualifications are right, the client is diverted. In any case, assuming that the accreditations are erroneous, the code catches the IP address, area, geolocation, city, nation, locale, and association of the assailant. Then, the code takes screen captures of the aggressor's PC circumspectly, without the assailant knowing. It additionally catches a picture of the aggressor. This data is all then sent through Message and Gmail administrations, which permits the client to make a legitimate move against the assailant or compromise them.

Contrasted with before variants, our HoneyTrap code is vastly improved on the grounds that it is lightweight and occupies next to no room. It likewise acts rapidly, making it challenging for the aggressor to prevent their subtleties from being sent. Moreover, it is easy to introduce, pursuing it an optimal decision for those searching for a profoundly powerful and easy to understand HoneyPot arrangement.

HARDWARE:

We are basically preparing a software that can be applied to different websites to protect them from honeypot deception so the use of hardware is negligible. We just used a laptop to code and prepare such a software and show the demo.

6. Proposed System

6.1 Objective of the Project:

The objective of the Honeypot deception method for real-time intrusion detection and prevention is to create a trap that appears to be a legitimate target for attackers, but in reality, is designed to detect and prevent malicious activity. A Honeypot is a decoy system that is set up to attract attackers and provide an environment where their actions can be observed and analyzed. The goal is to learn about the tactics, techniques, and procedures (TTPs) of attackers in order to improve security defenses and prevent future attacks.

Honeypots are often used as a proactive security measure to detect and respond to attacks in real-time. They can be deployed on internal networks, external-facing systems, or in the cloud, and can be customized to simulate a wide range of services and applications. By mimicking legitimate systems, Honeypots can lure attackers into a controlled environment where their behavior can be monitored and analyzed without risking damage to the actual production systems.

The use of Honeypots can provide valuable insight into the threat landscape, help identify new attack vectors, and improve incident response capabilities. By deploying Honeypots, organizations can gain a better understanding of their own vulnerabilities and improve their overall security posture. Additionally, Honeypots can help organizations meet compliance requirements by providing a mechanism to detect and prevent unauthorized access to sensitive information.

6.2 Novelty of the Project:

The Honeypot Double dealing Technique for Continuous Interruption Discovery and Counteraction project has a few novel perspectives:

1. Honeypot Innovation: The task uses honeypot innovation to distinguish and forestall interruptions. Honeypots are imitation frameworks that are intended to seem as though genuine frameworks yet are phony. By conveying honeypots, the undertaking can draw assailants from genuine frameworks and catch significant data about their strategies, methods, and methodology (TTPs).

2. Continuous Interruption Recognition: The undertaking utilizes ongoing interruption location to recognize and answer assaults as they happen. This permits the undertaking to rapidly answer assaults and forestall further harm.

3. Face Acknowledgment: The task utilizes face acknowledgment innovation to catch pictures and recordings of gatecrashers. This innovation permits the venture to distinguish gatecrashers and possibly connect them to different assaults.

4. Message and Gmail Administrations Mix: The undertaking coordinates with Wire and Gmail administrations to advise security work force progressively when an interruption is identified. This permits security staff to rapidly answer assaults and make a suitable move.

5. Moderate Code: The undertaking utilizes moderate Python code which makes it lightweight, quick, and simple to introduce. This permits the task to run proficiently on low-asset frameworks, like Raspberry Pi, and makes it open to a more extensive scope of clients.

6.3 Features of the Project:

Here are some potential features of a project implementing a Honeypot Deception Method for Real Time Intrusion Detection and Prevention:

1. Honeypot deployment: The project should include the deployment of one or more Honeypots in strategic locations to attract potential attackers.

2. Simulation of various services and applications: The Honeypots should simulate various services and applications that are likely to be targeted by attackers, such as web servers, email servers, and file servers.

3. Real-time monitoring: The Honeypots should be monitored in real-time for any suspicious activity, including attempts to exploit vulnerabilities or access sensitive information.

4. Notification and alerting: The project should include a mechanism for notifying security personnel when an attack is detected, and providing relevant information about the attacker's behaviour.

5. Analysis of attacker behaviour: The project should include tools for analysing the behaviour of attackers, including their tactics, techniques, and procedures (TTPs).

6. Integration with existing security tools: The project should integrate with existing security tools, such as intrusion detection systems (IDS) and security information and event management (SIEM) platforms, to enhance overall security posture.

7. Customization and scalability: The project should be customizable to meet the specific needs of

the organization, and scalable to accommodate future growth.

8. Reporting and analytics: The project should provide comprehensive reporting and analytics capabilities to help organizations understand the threat landscape, identify trends, and make data-driven security decisions.

Overall, the goal of a project implementing a Honeypot Deception Method for Real Time Intrusion Detection and Prevention is to provide an additional layer of security to detect and prevent attacks in real-time, and to gain valuable insights into the tactics of potential attackers.

6.4 Work Flow Diagram

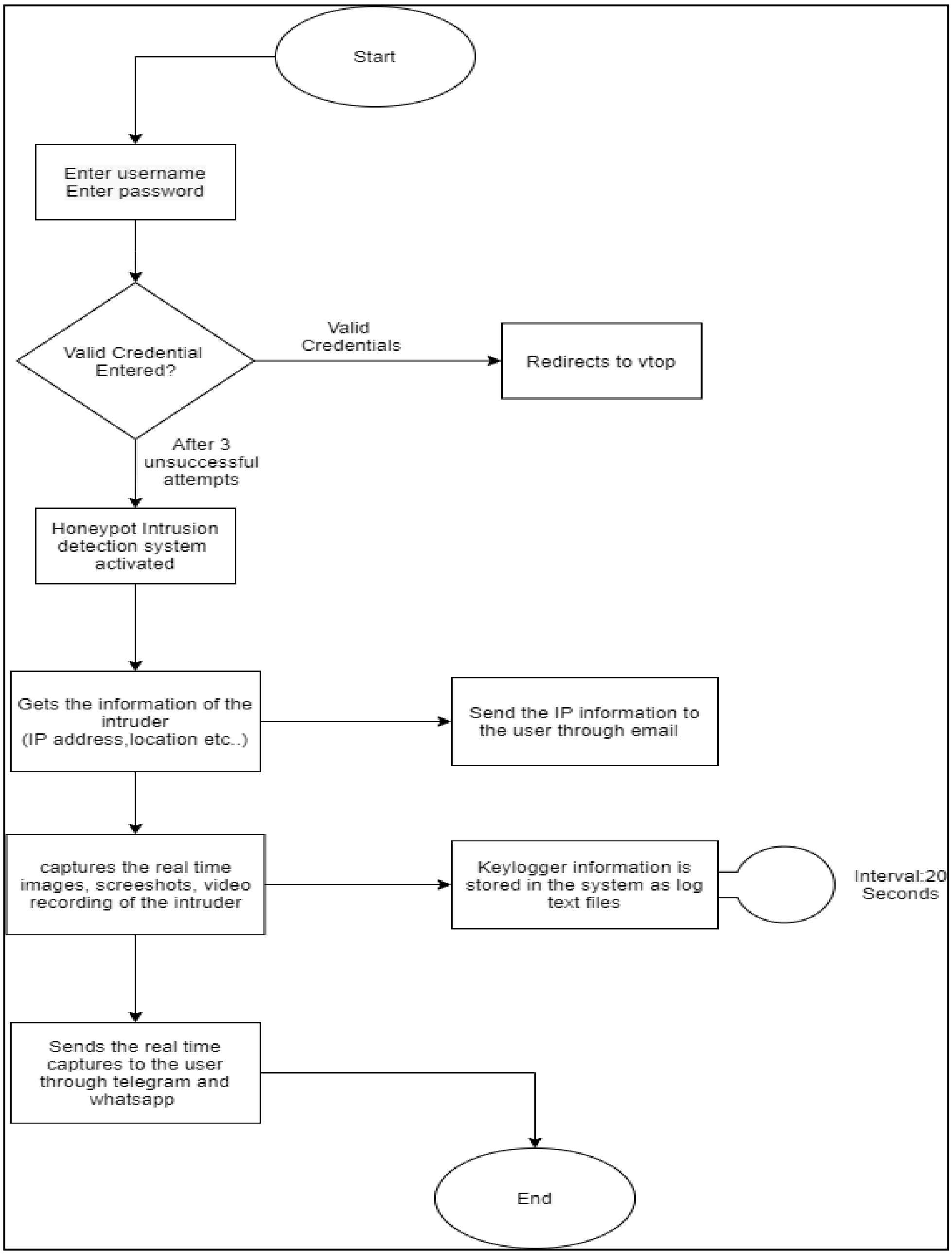

Start
Enter username
Enter password
Valid Credential Entered?
Valid Credentials
Redirects to vtop
After 3 unsuccessful attempts
Honeypot Intrusion detection system activated
Gets the information of the intruder (IP address,location etc..)
Send the IP information to the user through email
captures the real time images, screeshots, video recording of the intruder
Keylogger information is stored in the system as log text files
Interval:20 Seconds
Sends the real time captures to the user through telegram and whatsapp
End

Figure 2 – Flow chart of Honeypot.

6.5 Modules and their Description

Regardless, our association will make a site that includes a login entryway where our HoneyPot innovation will be incorporated. At first, the client will include their login certifications, for example, their username and secret word, through a Python script called Hash-change.

The login entrance will work in the accompanying way:

When the client signs in through the web server, the framework will check the accreditations that were placed through the Python script.

If the login qualifications are right, the client will be viewed as real and will be coordinated to their own VTOP account.

Nonetheless, in the event that the client enters mistaken certifications multiple times, the framework will start the backend location system to distinguish the gatecrasher.

This identification instrument includes following and putting away the IP address, area, and other urgent data of the assailant's framework.

We have led different trials to advise the client about the exercises regarding the gatecrasher or assailant.

With respect to enter:

The Python content will acknowledge login accreditations, for example, the username and secret phrase of the approved client.

For the sake of security, the backend will change over the secret phrase into a hashed secret phrase.

This information will then, at that point, be put away in a JSON document containing the username and hashed secret phrase of the approved client, which will be utilized to sign in to the web application.

The login entrance of the association's site will confirm the client's certifications and decide if they are authentic clients or aggressors/interlopers.

Regarding yield:

If the client's qualifications are right, they will be diverted to their VTOP entry after effectively signing in.

In the event that a gatecrasher enters mistaken certifications multiple times, the framework will create the accompanying result:

The gatecrasher's framework data, for example, their IP address, area (arranges, city, country), district, hostname, and other pertinent subtleties.

Ongoing pictures and video catches of the gatecrasher's face will likewise be taken.

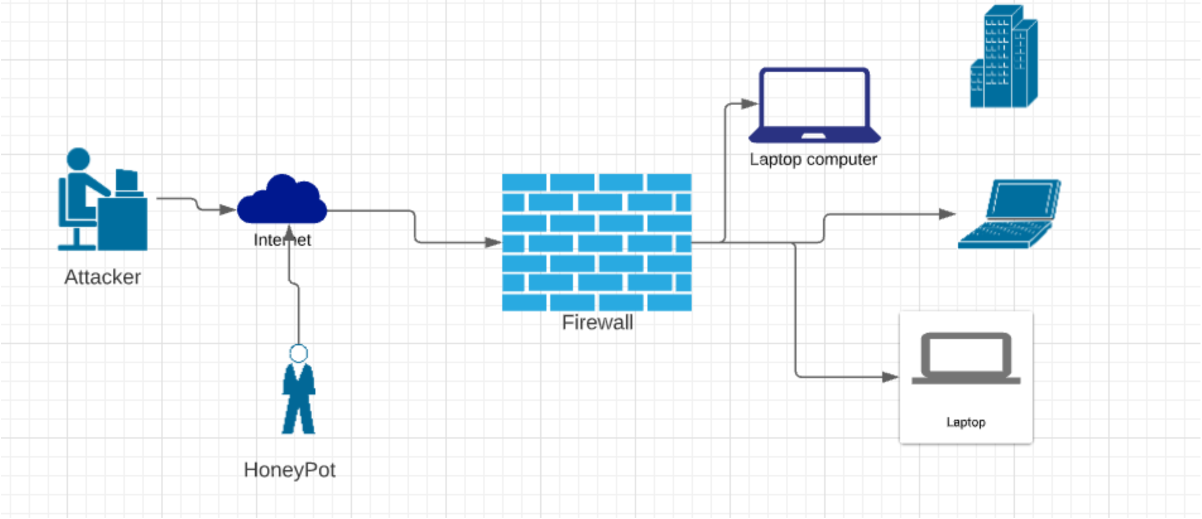
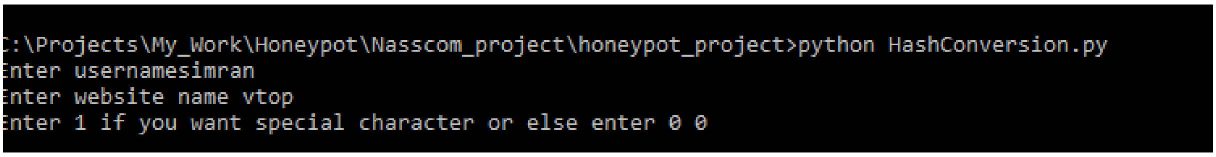


Figure 3 - Architecture Diagram of Honeypot.

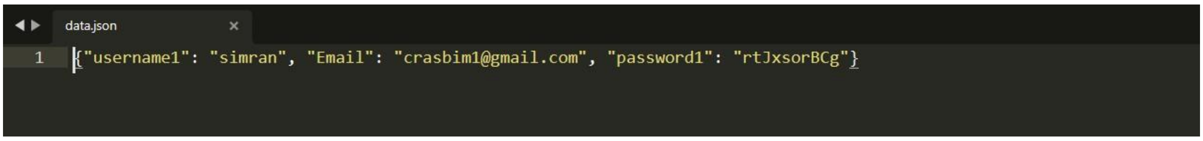
7. **Screenshot with explanation of each window**



```
:\Projects\My_Work\Honeypot\Nasscom_project\honeypot_project>python HashConversion.py
nter usernamesimran
nter website name vtop
nter 1 if you want special character or else enter 0 0
```

Figure 4 – Creation of Username and Password.

The process starts with entering seed hash values that are going to be used by Honeypot for authentication.



```
{"username1": "simran", "Email": "crasbim1@gmail.com", "password1": "rtJxsorBCg"}
```

Figure 5 –Data Stored in JSON files

Data is stored in json format as shown in figure.

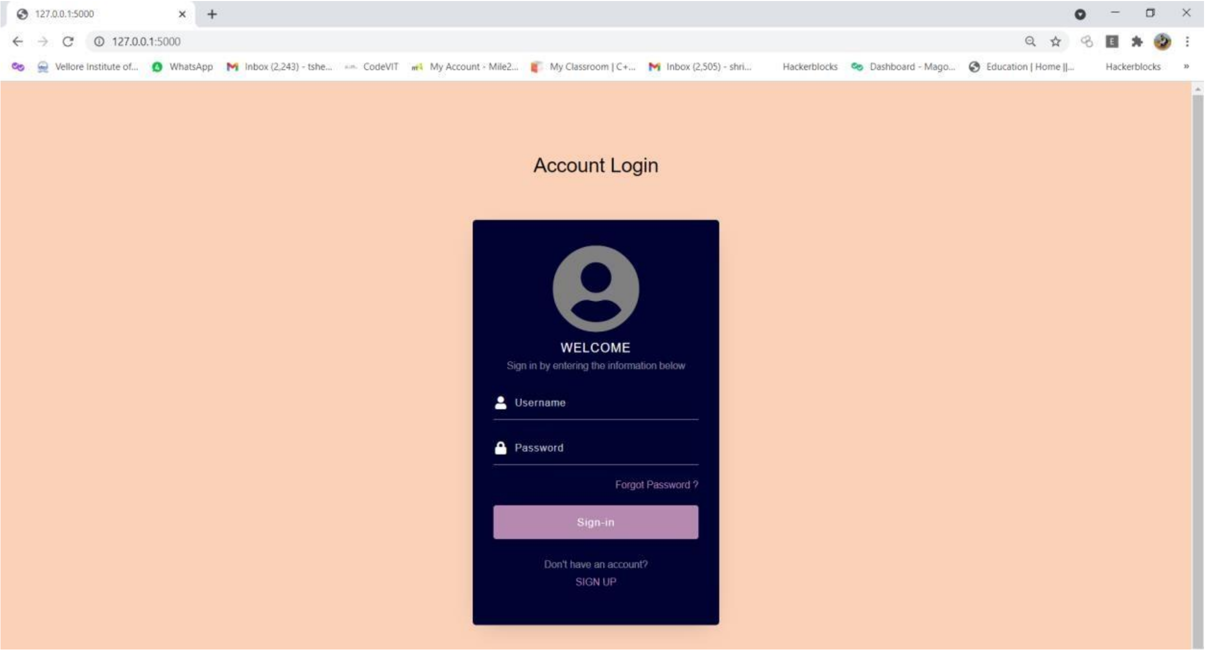
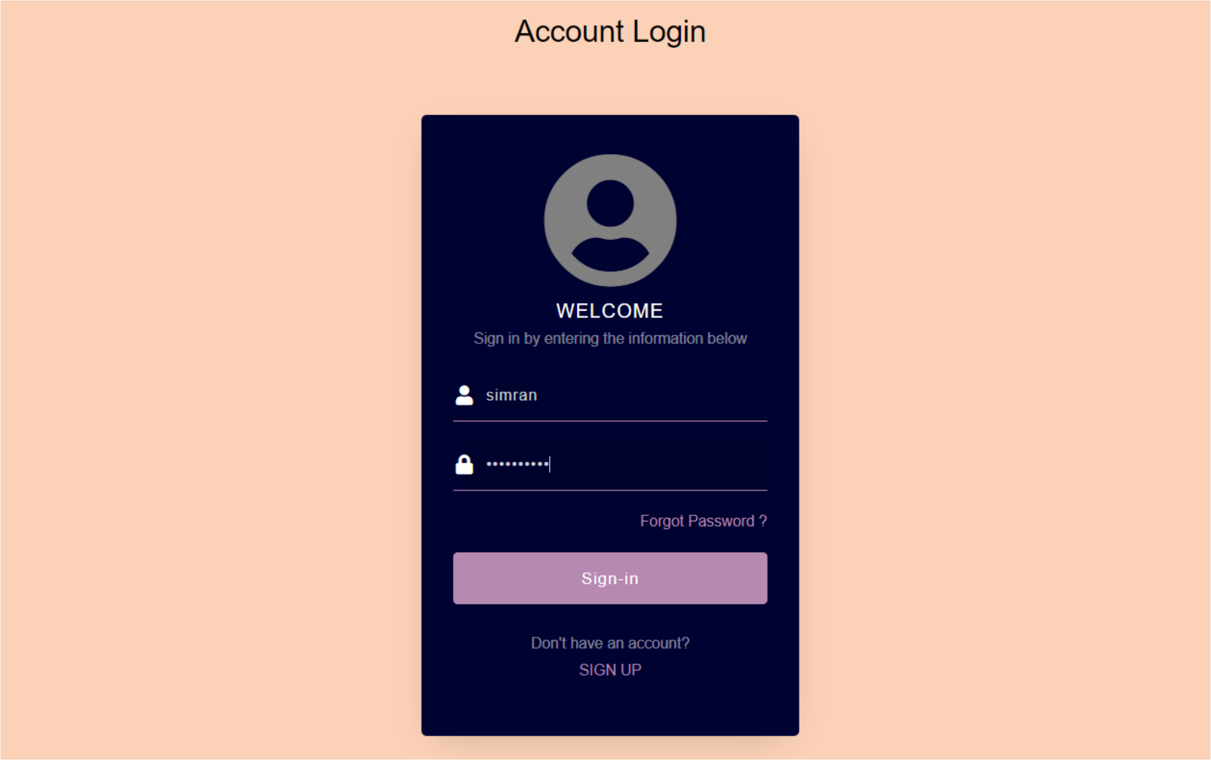


Figure 6 - Interface of Login Page.

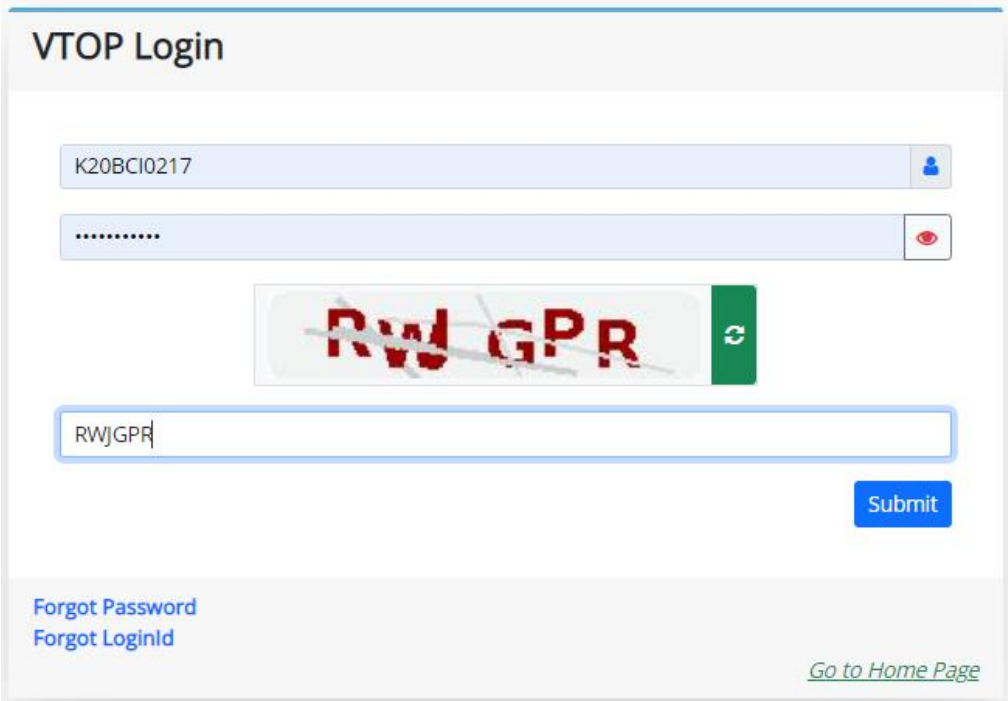
This is the sample authentication screen. The user enters username and password over here and accordingly is directed to different pages



simran

••••••••••

Figure 7 – Entering Login Credentials.



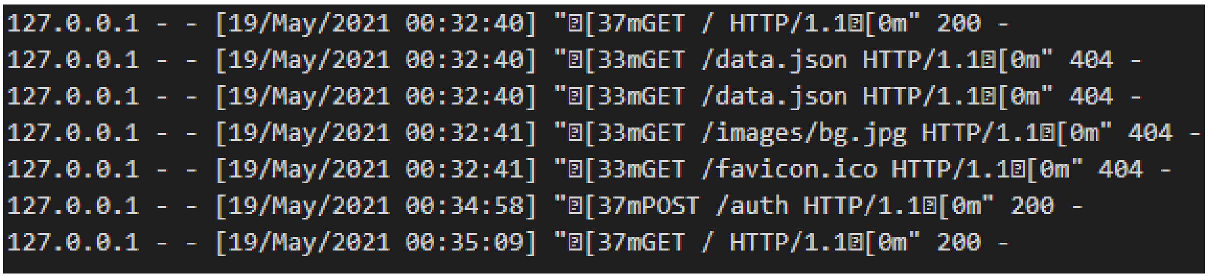
VTOP Login

K20BCI0217

•••••••••••

RWJGPR

Figure 8 – Correct credentials are entered will redirect to **Vtop** login page.

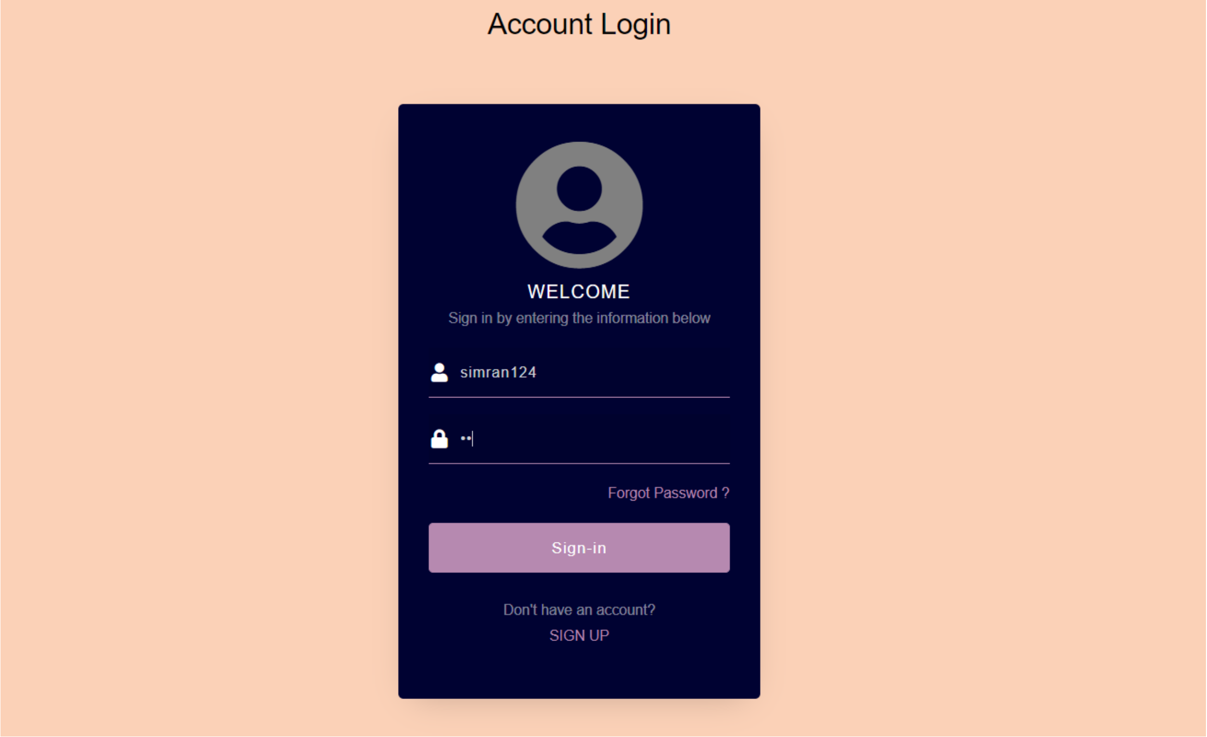


```
127.0.0.1 - - [19/May/2021 00:32:40] "□[37mGET / HTTP/1.1□[0m" 200 -
127.0.0.1 - - [19/May/2021 00:32:40] "□[33mGET /data.json HTTP/1.1□[0m" 404 -
127.0.0.1 - - [19/May/2021 00:32:40] "□[33mGET /data.json HTTP/1.1□[0m" 404 -
127.0.0.1 - - [19/May/2021 00:32:41] "□[33mGET /images/bg.jpg HTTP/1.1□[0m" 404 -
127.0.0.1 - - [19/May/2021 00:32:41] "□[33mGET /favicon.ico HTTP/1.1□[0m" 404 -
127.0.0.1 - - [19/May/2021 00:34:58] "□[37mPOST /auth HTTP/1.1□[0m" 200 -
127.0.0.1 - - [19/May/2021 00:35:09] "□[37mGET / HTTP/1.1□[0m" 200 -
```

Figure 9 – Backend server of Flask.

The flask backend verifies the password and compares it with the entered password. It also compares the username with the username given by the user.

If the correct username and password is entered they are automatically redirected to a success page. IN this case we redirect the user to the Vtop page. The end screenshot shows the 200-success given by the backend API for entering the correct username and password.



Account Login

WELCOME

Sign in by entering the information below

simran124

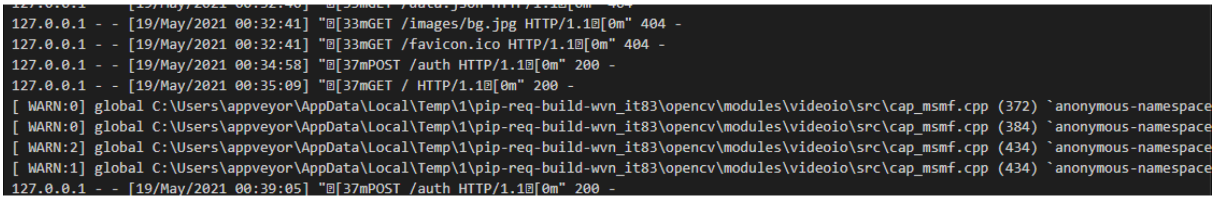
••

Forgot Password ?

Don't have an account?

SIGN UP

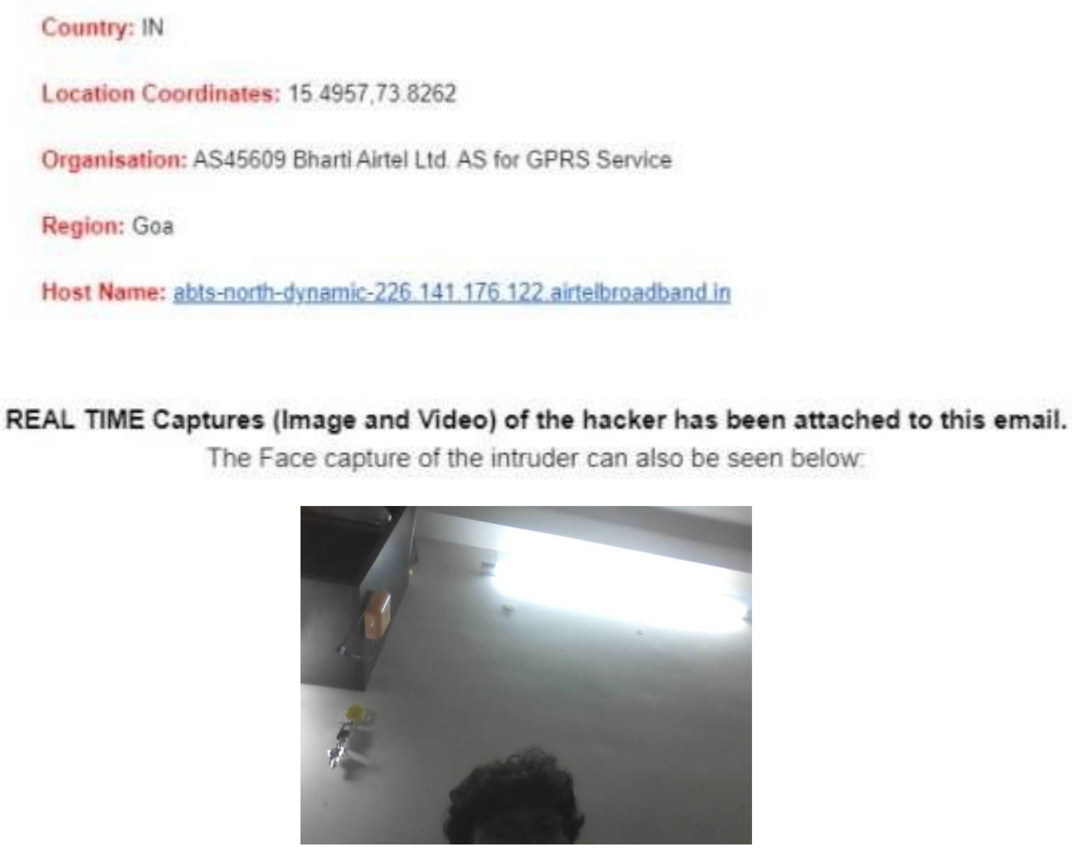
Figure 10 – Entering False Credentials.



```
127.0.0.1 - - [19/May/2021 00:32:40] "[33mGET /data.json HTTP/1.1[0m" 404 -
127.0.0.1 - - [19/May/2021 00:32:41] "[33mGET /images/bg.jpg HTTP/1.1[0m" 404 -
127.0.0.1 - - [19/May/2021 00:32:41] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -
127.0.0.1 - - [19/May/2021 00:34:58] "[37mPOST /auth HTTP/1.1[0m" 200 -
127.0.0.1 - - [19/May/2021 00:35:09] "[37mGET / HTTP/1.1[0m" 200 -
[ WARN:0] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-wvn_it83\opencv\modules\videoio\src\cap_msmf.cpp (372) `anonymous-namespace
[ WARN:0] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-wvn_it83\opencv\modules\videoio\src\cap_msmf.cpp (384) `anonymous-namespace
[ WARN:2] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-wvn_it83\opencv\modules\videoio\src\cap_msmf.cpp (434) `anonymous-namespace
[ WARN:1] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-build-wvn_it83\opencv\modules\videoio\src\cap_msmf.cpp (434) `anonymous-namespace
127.0.0.1 - - [19/May/2021 00:39:05] "[37mPOST /auth HTTP/1.1[0m" 200 -
```

Figure 11 – Flask server shows the wrong credentials.

When attacker attempts 3 wrong passwords then ours codes runs in back-ground and captures Screen shot and video of 10 sec and Attacker details like IP address, location, region and it send to user’s mail ID and also start taking the screenshot od screen and send it on telegram.



Country: IN

Location Coordinates: 15.4957,73.8262

Organisation: AS45609 Bharti Airtel Ltd. AS for GPRS Service

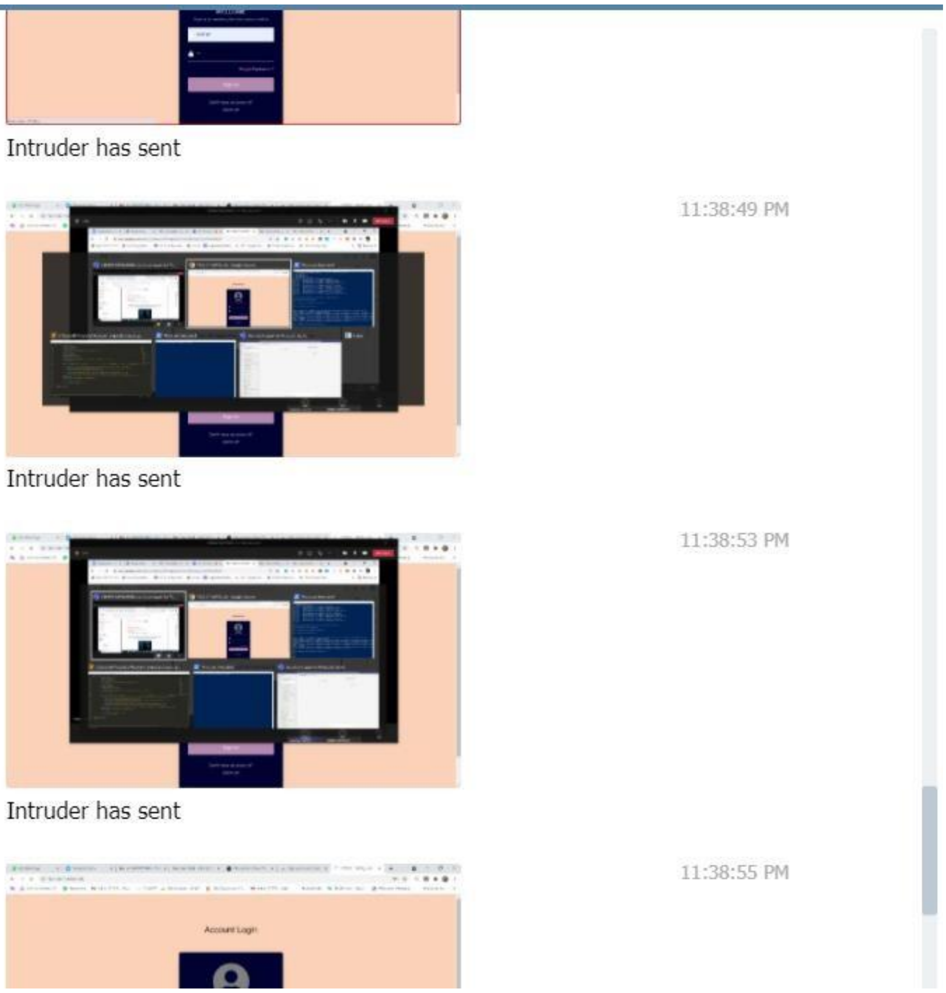
Region: Goa

Host Name: abts-north-dynamic-226.141.176.122.airtelbroadband.in

REAL TIME Captures (Image and Video) of the hacker has been attached to this email.

The Face capture of the intruder can also be seen below:

Figure 12 – Mail received to the User.



Intruder has sent

11:38:49 PM

Intruder has sent

11:38:53 PM

Intruder has sent

11:38:55 PM

Figure 13 – Screenshots of Attacks Desktop sent on Telegram.

In case an attacker is sitting on the system and tries to enter the wrong username and password, his photo, screenshots of his laptop and his details are sent to the owner. The screenshots also include the backend first indicating 404 since it's not able to serve the webpage because of the invalid credentials and then sending mail and telegram messages to the client

8.Conclusion

In this project we implemented a very strong system in the point of cyber security. We implemented a **honeypot-based mechanism** in which we didn’t just focus on recording the IP address, but we also worked on the **real-time implementation** by recording the **real-time screen shots** of the suspect system, and **face captures** of the suspected attacker. If we find any suspicious activity like not able to login in the portal for more than 3 times then the honeypot system will be activated in the backend which will start taking the screenshots of the intruder’s system and **alert the admins** via 2 important platforms: **Telegram and Email**.

These images, screenshots and logs can be studied later to know the exact activity of the user. If the consumer is discovered to be a criminal, legal action will be taken against them. The need to strengthen network security has increased in recent years. Honeypots can be used to achieve this level of protection. They are extremely useful as **counter-measures** from intruders’ attacks on systems. Such that security experts and analysts can identify who they are dealing with and ensure that network security is still maintained despite the rapid changes in network attacks. However, if the attackers are aware of such a device or are able to circumvent it, the whole process is rendered useless. As a result, this fact must be taken into account when designing a honeypot such that the intruder believes it is a genuine device rather than a trap.

9. References

1. Provos, N. (2004, August). A Virtual Honeypot Framework. In USENIX Security Symposium (Vol. 173, No. 2004, pp. 1-14).

2. Krawetz, N. (2004). Anti-honeypot technology. IEEE Security & Privacy, 2(1), 76-79.

3. Naik, N., Jenkins, P., Savage, N., & Yang, L. (2021). A computational intelligence enabled honeypot for chasing ghosts in the wires. Complex & Intelligent Systems, 7(1),477-494

4. Nazario, J.(2009). PhoneyC: A Virtual Client Honeypot. LEET, 9, 911-919.

5. Seifert, C., Welch, I., & Komisarczuk, P. (2007). Honeyc-the low-interaction client honeypot. Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 6.

6. Mairh, A., Barik, D., Verma, K., & Jena, D. (2011, February). Honeypot in network security: a survey. In Proceedings of the 2011 international conference on communication, computing & security (pp. 600-605).

7. Zhang, F., Zhou, S., Qin, Z., & Liu, J. (2003, August). Honeypot: a supplemented active defense system for network security. In Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies (pp.231-235). IEEE

8. Alata, E., Nicomette, V., Kaâniche, M., Dacier, M., & Herrb, M. (2006, October). Lessons learned from the deployment of a high-interaction honeypot. In 2006 Sixth European Dependable Computing Conference (pp.39-46). IEEE

9. Kuwatly, I., Sraj, M., Al Masri, Z., & Artail, H. (2004, July). A dynamic honeypot design for intrusion detection. In The IEEE/ACS International Conference onPervasiveServices, 2004. ICPS 2004. Proceedings. (pp. 95-104). IEEE.

10. Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., & Rossow, C. (2016). IoTPOT: A novel honeypot for revealing current IoT threats. Journal of Information Processing, 24(3), 522-533.