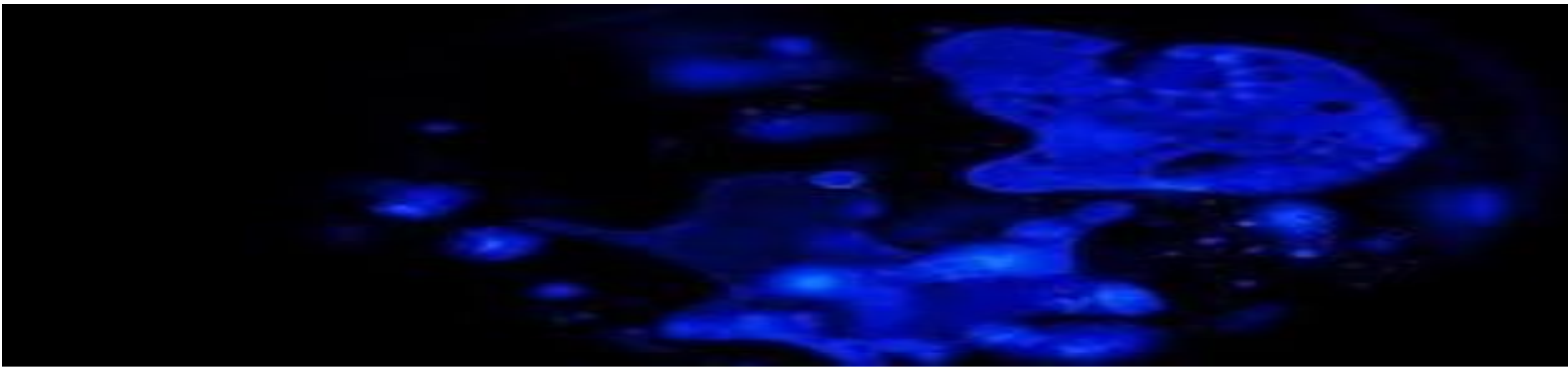


Inquirix

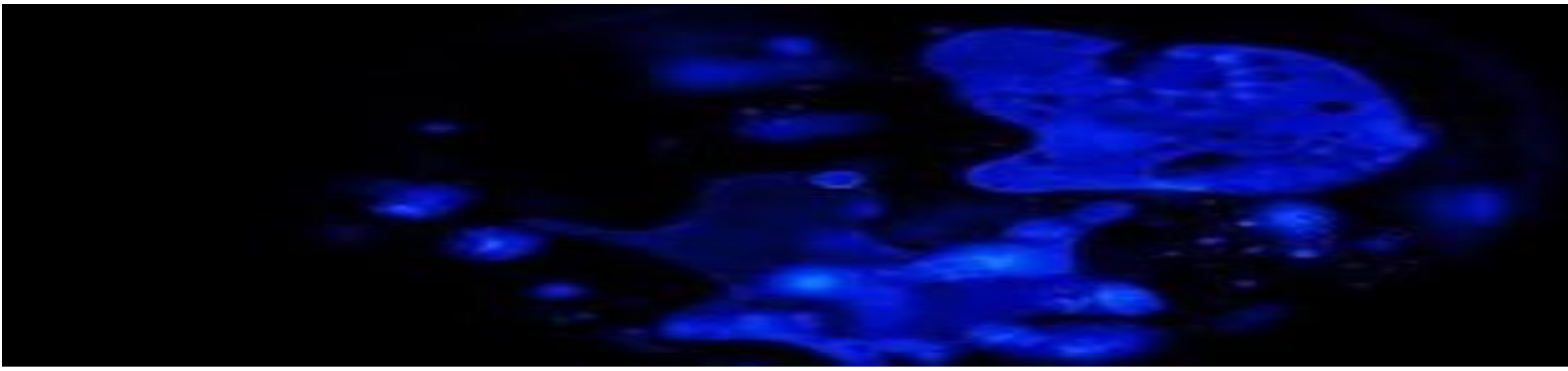
Messenger Bots –utilisation and threat protection

AUTHOR: Abi Waddell

http://www.inquirix.org

VERSION 1.0

May 2019



This gives a summary overview on bots which are used for messenger services, the security threats they pose and how such threats can be mitigated.

Firstly, a quick overview on bot types. 52% web traffic is due to bot activity, and this is made up of:

- Crawlers which visit accessible websites and their associated directories to look for new content and make records of such indexing to search engines.

- Data harvesting bots which scrape specific sites for all accessible data – usually this is for marketing and analytics

- Spam and marketing bots which detect user actions and behaviours to target specific marketing campaigns, or they might encourage users to divulge personal details such as email, sign up for products and special offers, partaking of surveys etc

- Social media bots which produce automatic content on social media sites among other actions.

- Malicious bots which seek to compromise user access and privilege or server and application functioning. There are sometimes several of these which form a network of bots which are controlled by a single master application.

- Game bots allow gamers to play against the application especially in the absence of other human players or allow the human player to control players which need to perform repetitive or automatic tasks.

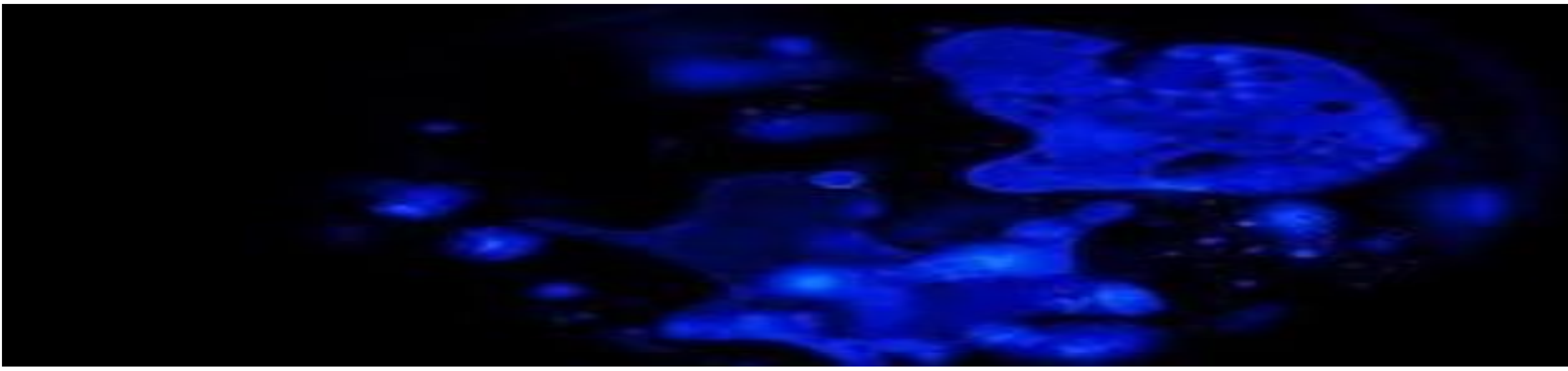
- And finally, there's chat or messenger bots which allow users to directly request actions to be performed or information in place of speaking to a human.

It's worth just mentioning that bots are sometimes confused with macros. Macros use a series of tasks that are performed after being triggered by the user; a bot however can react to its environment and does not necessarily need to be triggered manually.

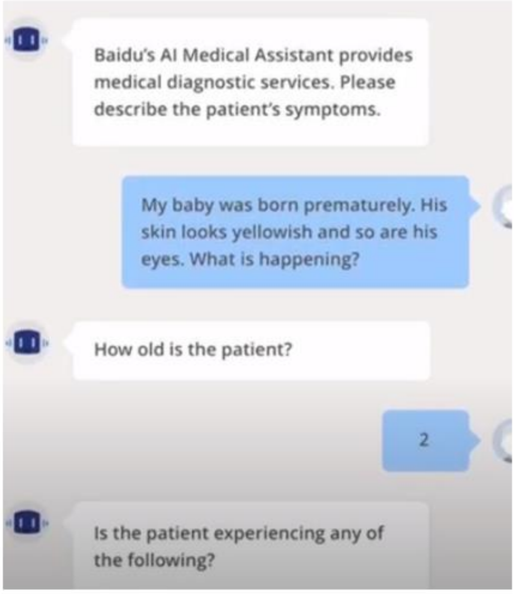
And now to focus on chat bots. Here are some of the main features and types:

- Support bots act like a `virtual assistant' and perform transactions or provide information, sometimes redirecting requests to human support staff.

- Personal assistant or `action' bots such as Alexa, Siri perform specific tasks – sometimes interfacing with other devices and hardware.



Chat bots work from a predefined `script' or flowchart where queries are directed according to the keywords used. The more advanced bots use artificial intelligence with learning capabilities for instance remembering past users or conversations from which to base a relevant response. Chat bots accept input either from text input or speech recognition.



Baidu's AI Medical Assistant provides medical diagnostic services. Please describe the patient's symptoms.

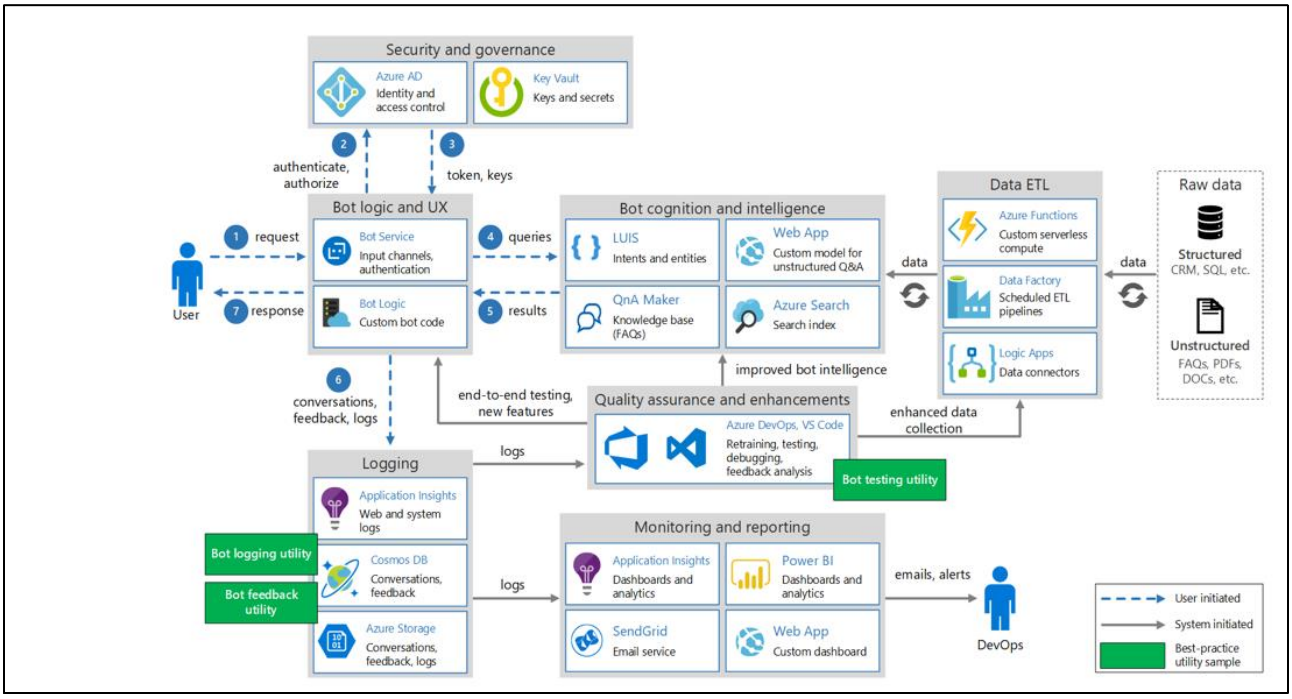
My baby was born prematurely. His skin looks yellowish and so are his eyes. What is happening?

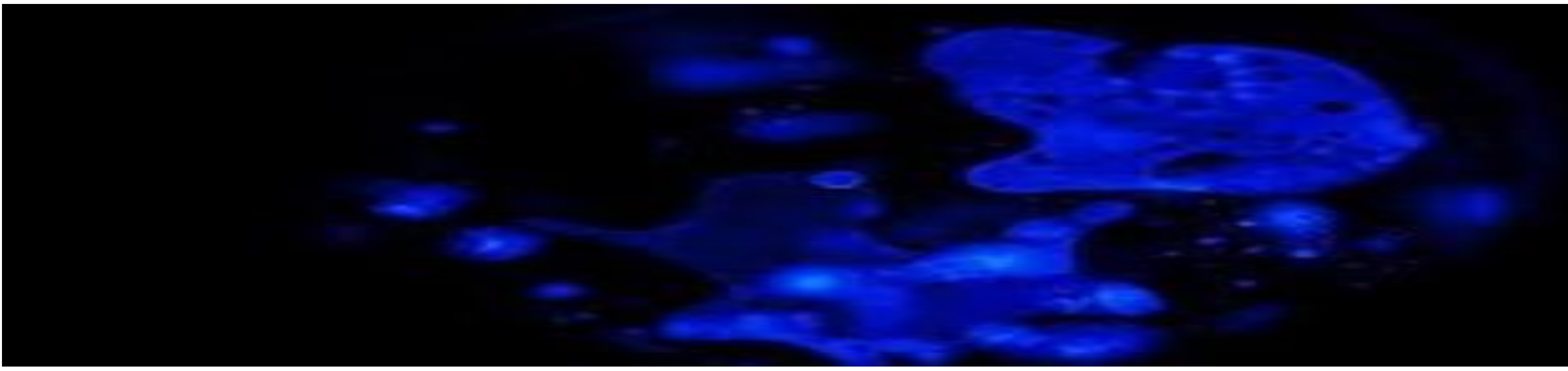
How old is the patient?

2

Is the patient experiencing any of the following?

This is a description of the infrastructure that the Azure bot uses:





This has been used as an example of the architecture of a typical bot – containing the elements common to many other bot platforms. There is a user interface – usually via the browser/web application front end, which then interfaces with the security mechanism that validates the user. User queries are passed via this front end to the language database which then forms queries to fetch information or perform actions using any backend databases. Adjoined to this process are logging, monitoring and quality control functions.

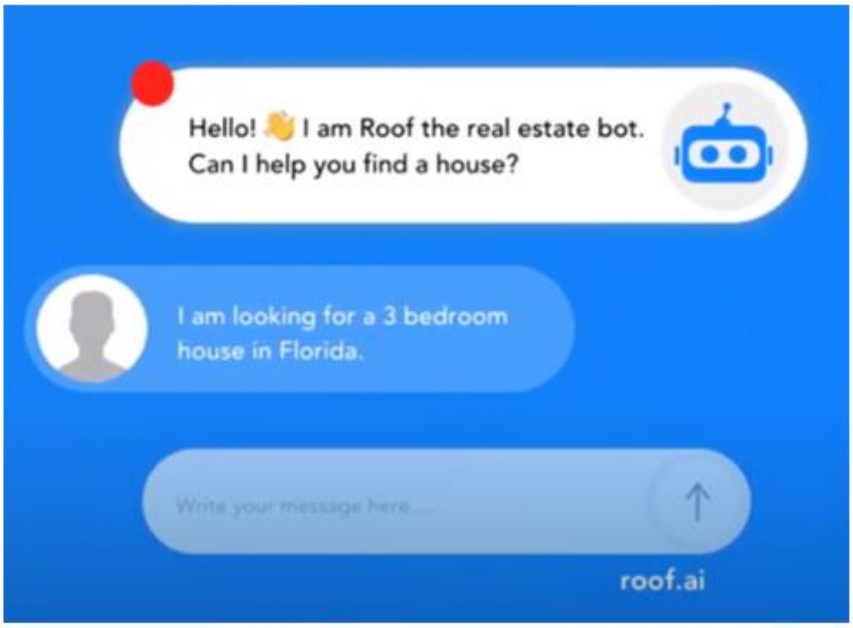
There are many benefits and use cases of chat bots, the most obvious one is that a bot can perform faster query response and more efficient actions than a human and can serve many more customers at once and at all hours of the day and night.

With quicker and more efficient service, customers may experience better satisfaction with the site, and may have more relevant content delivered to them based on their site behaviours.

Some bots encourage customers who are thinking of not buying a product (for example by abandoning a shopping cart) or leaving a service to find alternative products or find answers to their questions.

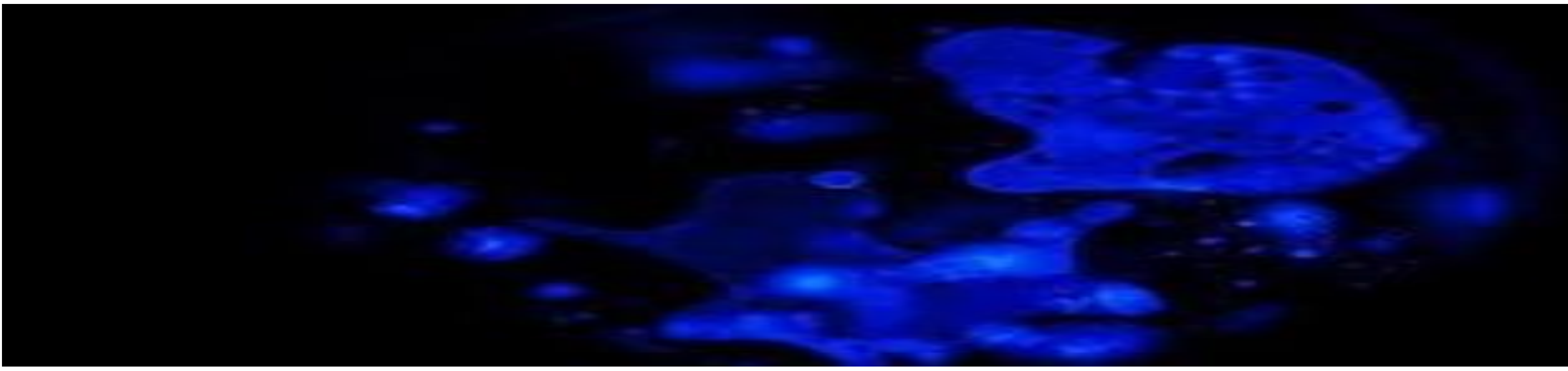
Chat bots are cheaper to run than people who require salaries, and they happily perform mundane, repetitive tasks that humans would not want to do. The customer is better understood because of intelligence gained about their previous or current behaviours.

Overall, the customer experience is improved with bots providing a faster, more efficient and customer-relevant service 24 hours a day and 7 days a week. The service or product provider gains by having increased sales from targeted customer marketing including the ability to make product or service recommendations, improved customer retention and a more efficient customer experience.



Hello! 👋 I am Roof the real estate bot.
Can I help you find a house?

I am looking for a 3 bedroom house in Florida.



A less common but relevant use of chat bots is in the provision of conversation and company to those who want to chat but who either don't want to or can't talk to a live person.

The main security risks posed by messenger bots are:

- Sensitive data disclosure – either of the host systems and network, the host application users and staff and customer data;

- Data corruption & theft – this is more likely to happen with command injection vulnerabilities and weak protections on any back-end databases and web servers;

- Denial of service which is usually caused by exploiting known system and application vulnerabilities.

This all leads to potential reputational damage, financial loss and competitor advantage.

Bot attacks take the same form as attacks on web applications in general. Some of the more common ones include:

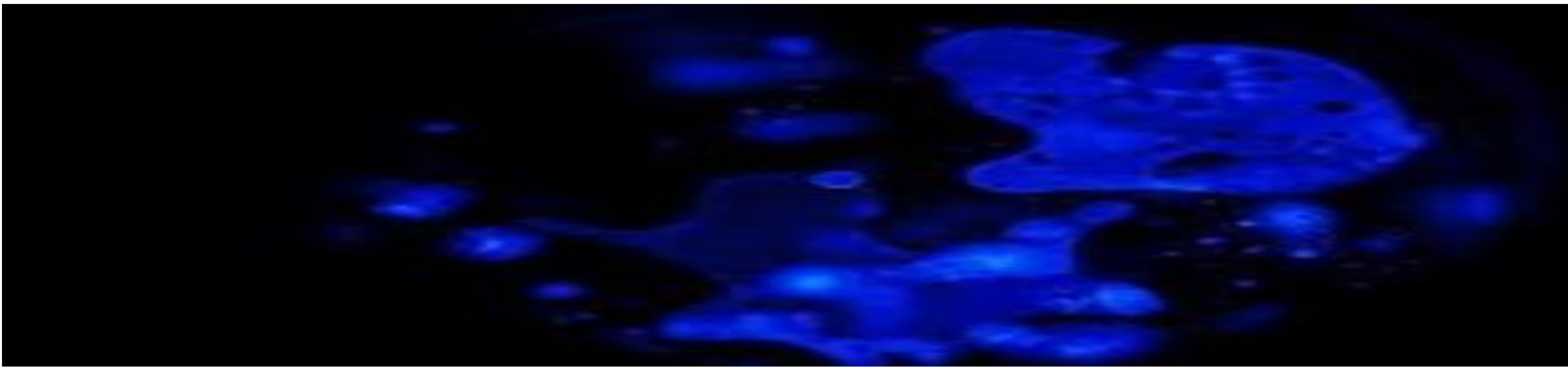
- Code injection in the chat window which can cause the bot to either disclose sensitive information, corrupt the application or make the bot or application behave erroneously. This image shows the Kuki chat bot responding to the command to show the contents of the current directory.

- Another attack is deceiving the bot into believing that the attacker is a legitimate user. The attacker would respond to any security or identity questions with either guessed or actual answers belonging to another user. Attackers may also impersonate a chat bot in order to communicate with users and obtain sensitive information from these users.

- Request flooding is a form of denial of service whereby an excess of chat requests causes the application to stop working properly.

- Cross-site scripting is a common web application attack which injects code into the application which is executed by the application and view by other clients. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access-privileges to sensitive page content, to session cookies, and to a variety of other information maintained by the browser on behalf of the user.

- User and sensitive files enumeration can be done via the login mechanism – often error messages provided by the application when a username is entered shows whether the name is valid or invalid; it can also be done via spidering and guessing the web directories for sensitive content; via social engineering of the chat bot and from open-source research of sensitive content which may be on other sites or accessible to search engines.



- An attacker may access sensitive data through open-source research of the site either on the site itself or using other sites, spoofing a legitimate user or internal system which is trusted by the application's other system, using command injection on the application to access the back-end databases, and actually logging in to the application as another user.

- Another possible attack vector is any file upload area on the site – an attacker could upload a malicious file to engage in server or user attacks.

- If an attacker can establish the version and type of software in use and if this software has publicly known vulnerabilities, then they could use either canned exploits or custom-made exploits to cause remote code execution, denial of service or information disclosure.

- Sometimes it is possible to simply guess the login details of another user. Usually this is achieved by finding publicly leaked passwords and re-using these passwords. Sometimes the default administrative credentials of a system or application are left unchanged, so it's possible to login using these details.

- Fake news - attackers could spread malicious rumours via social media sites.

- Privilege escalation attacks usually occur when a user has logged in as a normal low-level user and then gains access to content (files or directories) or gains access to performing specific actions which their current privilege level does not allow.

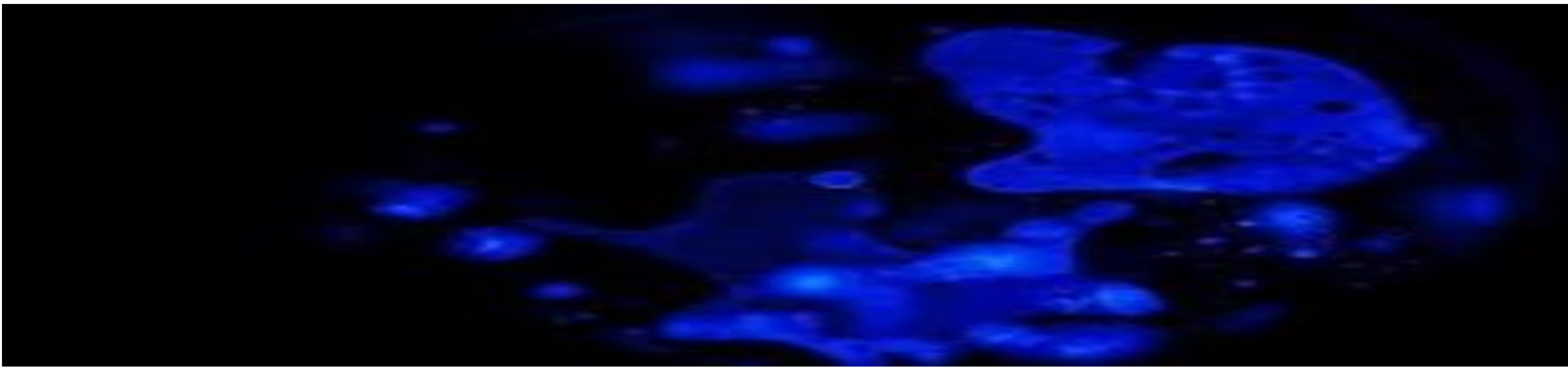
- And pivoting to internal systems can happen if the application has poor network controls and user validation along with poor content restriction.

Some of the best ways to mitigate these threats include:

- Ensuring that user sessions are encrypted as this will help to prevent any man-in-the-middle attacks;

- Using strong user authentication prior to any chat session. Strong user authentication typically uses two step or two factor authentication. Two step authentication requires the user to go through two sets of question and answers – for instance password followed by a pin number or memorable word. Two factor authentication requires the user to verify themselves via an additional mechanism – e.g., password followed by a code verification which has been sent to their phone.

- Sessions can be made more secure with the addition of non-cacheable, non-reusable cookies which have a set expiry time. Concurrent logins from the same user at the same time but from different devices or browsers can also be prevented.



- Web application APIs can be made more secure with the use of one-way asymmetric encryption for any passwords that are used, using strong authentication that only allows access to specific resources and can be expired after a set time, using rate limiting to prevent denial of service and user input validation so that script and other user behaviour which could cause unexpected system responses cannot happen.

- Applications should be hardened against potential or actual known vulnerabilities. This is usually achieved by ensuring the latest security updates and patches are applied, unnecessary ports closed and if possible, applications are tested on a regular basis. A patch management policy should be in place to ensure patches are applied in a timely manner. Attackers will be very good at finding out of date applications and servers and exploiting them where possible.

- The applications and servers should enable logging and should keep logs for the previous 6 months or longer. Useful actions to log are failed logins, successful administrative logins, application URLs visited and incoming source IP address.

- Applications should allow-list user input to prevent malicious command injection. In this respect only commands and characters which have been specified by the administrators are permitted. This is better than deny-listing which has a higher chance of allowing command injection.

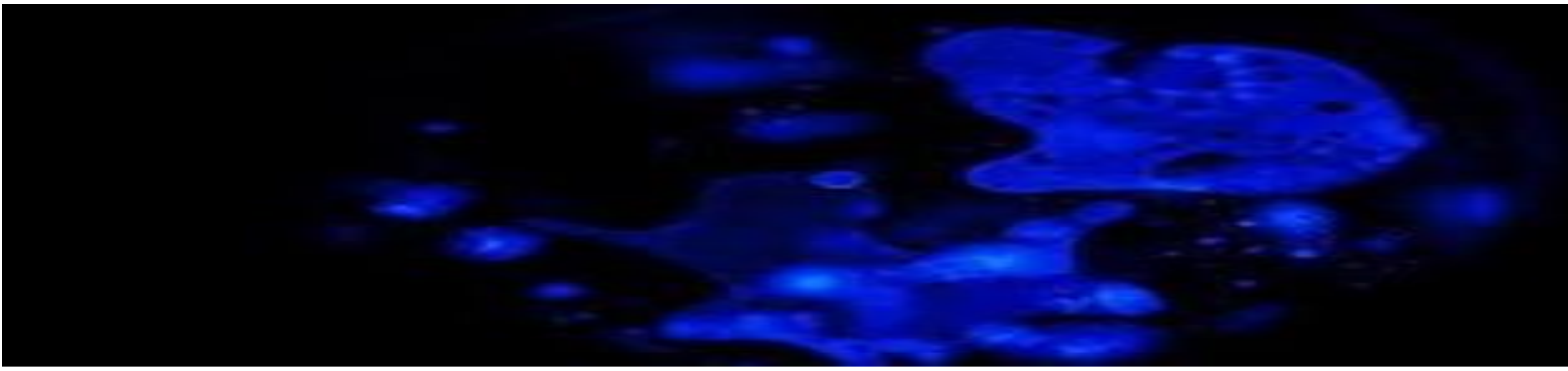
- Network controls include intrusion detection and firewalls which can block or rate limit unwanted traffic, both at the externally facing side of the network and on the back-end systems. Host detection and protection software can also provide an extra layer of defence.

Finally, user behaviour analytics is gaining more popularity now as this has the potential to spot attacker behaviour before an attack happens. Machine and deep learning are used to model the behaviour of users on corporate networks and highlights anonymous behaviour that could be the sign of a cyberattack. Such tools take log and other data from a variety of corporate systems and network devices to build a picture of normal versus anomalous user behaviour.

Chat bots can be exploited to perform activities against a target of interest, in particular intelligence gathering. It may be possible to enter commands in the chat which will get the bot to reveal sensitive data such as getting the bot to reveal the internal system directory.

Bots can also be used to run attacks on their own web application/host servers, acting as an intermediary to run commands. What makes this attractive to attackers is that the bot acts as a channel which has direct access to the back-end servers. Example attacks include command injection, denial of service and information disclosure.

In order to exploit any vulnerabilities in the chat bot attackers will try to circumvent any defences, including masquerading as genuine users, changing the source IP address of the attack system, using



script commands which haven’t been blacklisted by the application’s defence controls and accessing the web application via non-standard ports.