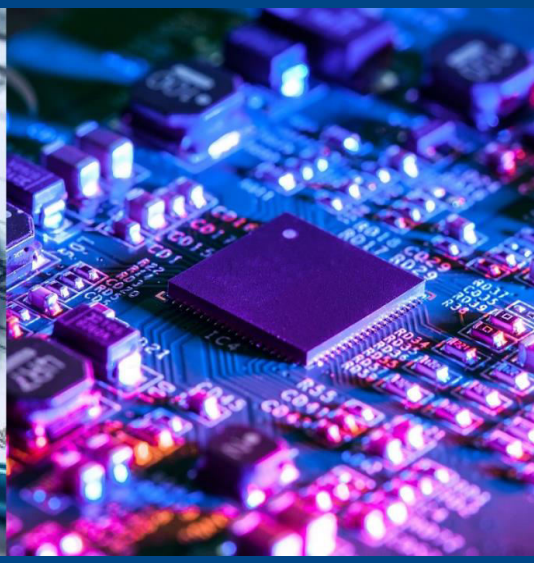
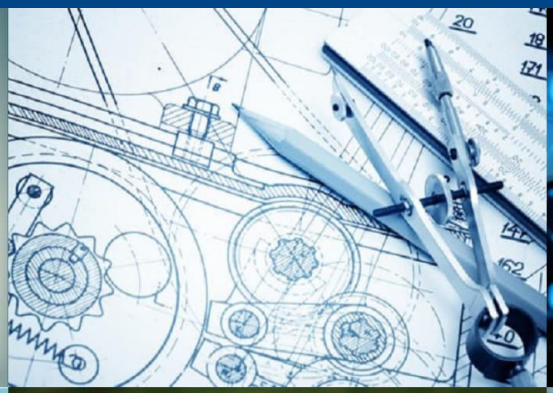
ISSN: 2582-7219

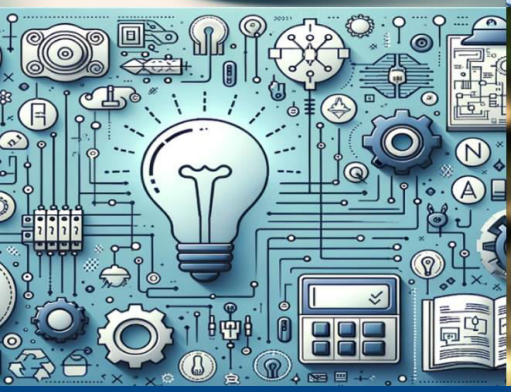




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 8, Issue 2, February 2025**

© 2025 IJMRSET | Volume 8, Issue 2, February 2025| DOI:10.15680/IJMRSET.2025.0802085

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI Driven ChatOps for DevSecOps: Automating Security Incident Response

Balajee Asish Brahmandam

Independent Researcher, University of Texas at Austin, Austin, Texas, USA

ABSTRACT: Though security incident response sometimes trails in speed and integration, DevSecOps approaches hasten software delivery. This study offers a ChatOps platform driven by artificial intelligence to automate and simplify security incident response inside DevSecOps processes. We show how artificial intelligence (AI) combined with ChatOps (chat-centric operations) allows fast detection, collaborative investigation, and automated remediation of security problems in real time. The method puts machine learning-based threat detection into team chat systems (e.g. Slack, Microsoft Teams), hence enabling security bots to alert teams of incidents with enhanced context, help in triage, and carry out containment operations using chat commands. We assess the advantages of this integration, including quicker reaction times, better teamwork, less human error, and ongoing incident learning. Automated documentation in a banking industry case study shows notable response speed (minutes rather than hours) and compliance improvement. We talk about issues such tool integration, false positives, and confidence in automation and recommend future improvements such sophisticated language models for incident management. The findings show that ChatOps driven by artificial intelligence may turn incident response into a proactive, quick procedure in line with DevSecOps agility.

KEYWORDS: DevSecOps; ChatOps; Security Incident Response; Automation; AI in Cybersecurity; SIEM; SOAR

I. INTRODUCTION

Though incident response procedures still provide a weak link, modern companies have adopted DevSecOps to include security into the quick DevOps lifecycle. Often slow, reactive, and separated from development, traditional incident response struggles to keep up with continuous deployment speed. High numbers of security alerts cause security teams alert fatigue, which results in practical missing or delayed reactions. For instance, one bank's security department said that human analysis of every alarm was so time-consuming that it caused delayed reactions and higher exposure risk. Delays in detection and containment can be expensive in an age of strong cyber-attacks. Industry standards like NIST SP 800-61 emphasize that "rapidly detecting incidents" depends on an efficient incident management capacity and helps to reduce damage. Incident response, therefore, must be as quick and agile as DevOps.

By integrating operations and tools into a cooperative conversation space, ChatOps has become a hopeful model to close this gap. ChatOps—short for "chat operations" is a way of working in which team communication channels double as the interface for executing tools and scripts. Enabling discussions to be instantly transformed into action, "ChatOps is a collaboration model that connects people, tools, process, and automation into a transparent workflow," as one practitioner put it. All pertinent warnings, data, and remedial orders are funneled via a continuous chat room in a ChatOps-driven incident response. This strategy tightens decision-making feedback loops and eliminates silos by means of real-time visibility of the same information and context by everybody. During a crisis, teams no longer must toggle between distinct consoles, emails, and ticketing systems; the chat itself becomes the war room for handling the event. When dangers change at machine speed, though, just centralizing communication is insufficient. The amount and complexity of information could still overwhelm human responders.

Here is where artificial intelligence integration is vital. Improvements in security analytics and machine learning allow for quicker identification of abnormalities and risks than human analysts can handle. AI-driven security tools for example, IBM's Watson-powered security intelligence or Microsoft's cloud AI analytics can comb through records and discover suspect trends automatically. Infusing ChatOps with artificial intelligence will help us to move from a strictly reactive stance to a more proactive and automated incident response. In an AI-driven ChatOps system, smart bots constantly monitor systems, create alarms with rich context, and even start containment actions before a person is engaged. The cooperative conversation then lets people confirm, debate, and guide the activities of the artificial

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

intelligence. This combination of machine speed and human supervision promises to reduce response times from hours to minutes and to better match security response with the ongoing delivery attitude of DevSecOps.

We provide a thorough methodology for AI-Driven ChatOps in DevSecOps in this article and evaluate its efficacy in automating security incident response. Including important industry solutions and recommendations, Section II surveys enabling technology and related activities. The suggested architecture and methodology for combining artificial intelligence with ChatOps is covered in Section III. Results and advantages seen are covered in Section IV along with a financial sector case study showing actual impact. We critically assess the difficulties and constraints of AI-driven ChatOps in Section V and investigate possible future developments, including more sophisticated artificial intelligence assistants. Ultimately, Section VI finds that AI-driven ChatOps can greatly improve the speed and cooperation of incident response, therefore strengthening enterprises against attacks without compromising DevOps agility.

II. LITERATURE REVIEW

Both business and standards have long acknowledged the demand for fast, automated incident response in DevSecOps. Timely detection and reaction are stressed in the NIST Computer Security Incident Handling Guide (SP 800-61), which underlines that companies must act fast to limit harm. In conventional configurations, security operations centers often depend on Security Information and Event Management (SIEM) systems to gather warnings and on playbooks or runbooks for reaction. But if not included into the development process, these may be rather sluggish. DevSecOps, which seeks to integrate security into quick DevOps processes, demands that incident response be seamless and ongoing. This has sparked further curiosity in methods that include developers in the loop and automate incident management. Recent studies have started looking at incident management using AI-assisted cooperation. Reflecting increasing scholarly interest in this intersection of artificial intelligence, DevOps, and SecOps, Brooks et al. (2025) explore how AI-powered ChatOps might enhance incident resolution and cooperation efficiency in DevOps teams.

ChatOps as a concept was popularized in the mid-2010s, especially by GitHub and others, to expedite operations by using chat platforms. Atlassian and other DevOps tool makers have recorded the benefits of ChatOps in incident management, stressing how it maintains everyone on the same page and dismantles information silos. A ChatOps system allows team members to run commands through the chat interface while a bot or integration publishes alarms and pertinent data into a shared chat channel. That channel logs all activities and conversations, therefore producing a consistent timeline of the event. ChatOps promotes openness and shared situational knowledge during events by centralizing communication. Operations' early ChatOps adopters have noted quicker coordination and better team culture. Regarding security, "Security ChatOps" has been described as conversation-driven inquiry in which analysts, tools, and bots work together in one window. This idea has spawned solutions and plugins linking chat platforms to security systems, hence enabling analysts to ask questions and execute incident response activities straight from the conversation. For instance, the security team at Palo Alto Networks observed that using ChatOps might reduce response times by removing the need to move between several interfaces and increases transparency since every analyst can see the orders carried out and outcomes in real time. These market insights lay the groundwork for including artificial intelligence (AI) into the ChatOps loop.

Parallel to the development of ChatOps, Security Orchestration, Automation, and Response (SOAR) technologies have seen notable expansion. SOAR systems such as IBM Resilient, Splunk Phantom, and Palo Alto Cortex XSOAR let security teams automate playbooks by codifying incident response processes. Once specific criteria are satisfied, these systems can automatically carry out activities include alert enrichment, notification delivery, or endpoint quarantine. These features have been even more improved by artificial intelligence included into security monitoring. A remarkable example is IBM QRadar Advisor: IBM's QRadar SIEM combines the Watson artificial intelligence to examine threats and offer insights that would often need a seasoned analyst. By means of threat intelligence and previous data, Watson can automatically investigate an offense and generate a natural-language report with results and confidence scores. Such AI-driven analysis can then be pushed into ChatOps indeed, IBM has shown ChatOps interfaces where Watson's recommendations (e.g., identifying a possible root cause or recommending a remedy) are posted into a Slack channel for the analysts to consider. This illustrates how artificial intelligence may be a force multiplier, managing the great cognitive load of analysis and allowing people concentrate on decision-making.

Cloud companies have also included artificial intelligence in their security products. A cloud-native SIEM, Microsoft Sentinel (formerly Azure Sentinel) detects anomalies and threats using User and Entity Behavioral Analytics (UEBA)

© 2025 IJMRSET | Volume 8, Issue 2, February 2025| DOI:10.15680/IJMRSET.2025.0802085

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and built-in machine learning. One of Sentinel's main advantages is its close connection with automation processes: it enables Azure Logic Apps playbooks to activate incident responses. Microsoft's papers show situations in which a Sentinel rule identifying a high-priority incident will automatically send a message to a Microsoft Teams channel to notify the on-call analysts. From there, analysts can even authorize some automatic activities and communicate via Teams. Essentially, this puts a ChatOps model into effect: Sentinel (the artificial intelligence brain) finds a problem and alerts people via a chat tool (Teams), where they can guide the next actions. Other manufacturers have comparable interfaces; for example, Palo Alto Cortex XSOAR offers out-of-the-box Slack and Teams bots replicating its incident "war room." All activities synchronized back to the central SOAR console allow analysts to investigate and control incidents straight from Slack. This implies that the Slack bot could run a virus scan or retrieve firewall logs, and the findings would be delivered in chat. Such integrations show that ChatOps is not only a convenience but also a unified interface for organizing complicated, multi-system reactions using artificial intelligence and automation behind the scenes.

Though case studies and early adopters show proof of its efficacy, the idea of including artificial intelligence into ChatOps is still developing. Atlassian's incident management systems have included AI-assisted capabilities; for instance, they employ big language models to automatically summaries incident chat conversations and modify Jira tickets for post-incident review. This guarantees that knowledge is recorded and helps to lighten the paperwork load on teams. Reports have surfaced in the practitioner community, however, of utilizing sophisticated chatbots—some driven by GPT-style models—in security operations to respond to inquiries and direct analysts. The trend indicates that future SecOps teams might regularly collaborate with an "AI assistant" in their conversation who might clarify a warning or recommend a course of action. The DevSecOps community is beginning to view such AI-driven ChatOps as a recommended practice for agile security. All things considered, the research and business solutions agree on the following:

- Modern incident response depends on speed and cooperation.
- ChatOps enhances visibility and coordination; and
- AI and automation can greatly lower response and detection times.

Building on these ideas, the following part explains our approach for integrating these components into a consistent incident response system.

III. METHODOLOGY OF PROPOSED SURVEY

AI-Driven ChatOps for Incident Response, our suggested approach, combines artificial intelligence-based threat identification, cooperative communication, and automated action execution. The high-level architecture and workflow of the system is shown in Figure 1. The process can be summarized in five stages: from early discoveryto confinement and learning. We outline the technical integration strategy next, stage by stage, along with the enabling elements.

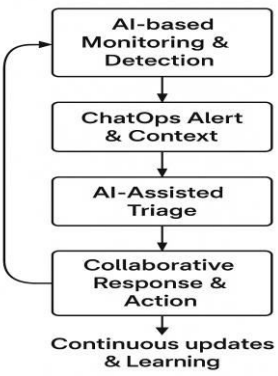


Figure 1: AI-driven ChatOps incident response workflow. AI-powered monitoring systems detect threats and send an alert with context to the ChatOps channel. The team (and AI bot) collaborate in chat to triage and decide on actions,

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

which are then executed via automation. Post-incident insights (dashed arrow) feed back into the AI models to improve future detection and response.

1. The pipeline starts with ongoing monitoring by AI-enabled security tools. The pipeline starts with ongoing monitoring by AI-enabled security tools. This covers intrusion detection systems, endpoint detection and response (EDR) tools, SIEM systems with machine learning analytics, and so on. These tools are set up to find abnormalities or recognized threat trends. For instance, an EDR might utilize behavioral analysis to catch malware whereas Microsoft Sentinel could apply anomaly detection models to logins and network activity. The system automatically creates an incident or alert when high confidence identification of a suspicious event or trend occurs. Importantly, the AI/analytics system notifies the ChatOps channel utilized by the DevSecOps team rather than just recording the occurrence. Practically speaking, integration hooks accomplish this: for instance, a Sentinel playbook or custom script delivers a message to a Slack/Teams channel via webhooks or bot APIs when an incident is created. Usually, the alert provided in chat is a summary of the problem and related background information. Automating the detection-to-notification process eliminates any delay between a security event happening and the team becoming aware of it.

2. Once an alert appears in the ChatOps channel such as a special "#security-incidents" Slack room or a Microsoft Teams incident chat the system offers additional context to help instant comprehension. Key information collected by artificial intelligence is also included by the ChatOps bot—or integrated notification system—not only the alarm (e.g., "Possible data exfiltration detected on Server X"). The message, for example, might contain a timestamp, the impacted host or application, the triggering anomaly (e.g., "outbound data spike of 500MB to unknown IP"), and a risk or priority score. The alert can relate to comparable events by integrating with knowledge bases and historical incident data, e.g., "Pattern resembles incident #248 (SQL injection attempt) last month." In certain versions, the bot may even include suggestions at this point. IBM's Watson-based systems illustrate this by automatically including a recommended diagnosis or action; for instance, Watson may deduce "suspected credential compromise" and recommend "reset user password and investigate login origin". The chat system notifies all channel team members @mentioned or otherwise, so guaranteeing that on-call respondents notice the alert right away (usually via mobile push notifications). A constant chat lets even later joiners scroll up to view the whole background and history of the alert.

3. The incident is now visible to the team, so triage which artificial intelligence helps to speed up is next. AI-assisted triage and investigation: The problem is now visible to the team, therefore triage which artificial intelligence help expedites is next. All in plain language or via specified instructions, team members can engage with the ChatOps bot to ask more information or do rapid checks. For instance, an analyst could say, "Bot, show related alerts in last 24h for Server X," and the SIEM-integrated bot would then be able to quickly fetch and display the pertinent data. This means the analyst doesn't have to manually search several systems. By offering smart help, machine learning enters the picture: the bot can be configured to grasp typical questions and retrieve responses using NLP (Natural Language Processing), so functioning as a security chatbot. Moreover, by linking the event with other signals, the artificial intelligence can give it top priority. Should several low-level alarms be connected, the artificial intelligence can underline the whole as a single high-priority event. At this point, artificial intelligence helps less seasoned team members by suggesting probable reasons or next actions depending on past trends. The bot, for example, could say, "This alert is probably a false positive, similar to events seen before, because X, Y, Z," or "This seems serious; 85% probability of ransomware infection affecting 3 hosts". Trained models using historical incidence data provide such advice. Focusing their attention correctly, this AI-driven triage enables the team to rapidly grasp the extent and influence.

4. Using the chat as the coordination tool, the human operators—which may include developers, ops engineers, and security analysts—collaboratively decide on containment or remedial steps with information at hand. In this phase, ChatOps excels by allowing the chat interface to initiate immediate action. The team can run commands by engaging with the bot via interfaces with orchestration tools as automation scripts or SOAR platforms. The channel, for instance, might offer an interactive prompt: "Quarantine Server X? Should a team member click "Yes" or enter an approval command, the bot will start a specified containment playbook based on the channel's interactive prompt: "Quarantine Server X? (Yes/No)." Chat instructions can initiate actions including host isolation from the network, user account deactivation, service restart, and patch deployment. Logged and secured,

© 2025 IJMRSET | Volume 8, Issue 2, February 2025| DOI:10.15680/IJMRSET.2025.0802085

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

these commands are only run by authorized users or roles. One implementation of this strategy showed an analyst could isolate an affected server with a single chat message using a Teams bot linked to an automation engine. Every conversation member instantly sees the operation being done—for example, the bot publishes "Executing containment: isolating DBServer12 from network". ChatOps's cooperative character allows for simultaneous discussion and input from others even when one team member initiates an action. The conversation is the virtual "war room"; there is no need to call a different phone. This not only guarantees everyone knows the most recent situation but also quickens decision making. During response, the AI still helps; for example, if the team is unclear about an action, they may request a risk assessment from the bot, and the AI could answer with probable results. Sometimes referred to as human-in-the-loop automation, this is the result of combining human judgment with artificial intelligence speed, which automates regular processes under human supervision for important judgments.

5. As the problem is being managed, the ChatOps bot keeps the channel updated on any automated tasks. The bot publishes the outcome (success/failure and any relevant information) if an action was carried out, such as banning an IP at the firewall. The complete history of events from the original alarm, team conversations, directives sent to the ultimate resolution is stored in the chat. Once the immediate danger is under control and handled, this transcript becomes a useful relic. For record-keeping, it can be automatically exported to a ticket or incident management system (for example, logged to ServiceNow or Jira). Many compliance rules call for a post-incident report; here the conversation log offers an exact timeline of who did what and when. Furthermore, the AI systems get this information back to enhance next performance. The features and results of the event can train machine learning models (with suitable data management). Should the AI suggest a beneficial action, that strengthens its learning; should it overlook an indicator humans caught, that knowledge may be included into next detection logic. The outcome is ongoing development: eventually, the AI gets more precise in seeing and diagnosing problems, and the playbooks may be honed depending on what the team discovered. Because all insights and choices are recorded in one location, the ChatOps method naturally encourages post-incident review. Team retrospectives can run the chat to find any delays or miscommunications. A feedback loop between human knowledge and artificial intelligence, this closed-loop learning is: every event managed by AI-ChatOps makes the system wiser (via ML training) and the team more informed (through collected lessons).

IV. RESULTS

Adopting the AI-driven ChatOps approach would help companies to see notable changes in general operational efficiency and important incident response statistics. The key benefits identified coincide with DevSecOps ideals of speed, collaboration, and continual improvement. Industry reports and our study provide the foundation for the following results:

Quicker Response Times: Automation significantly cuts the time from detection to first reaction. Chat delivers alerts to responders right away; containment measures (such as isolating a compromised server) can be started within seconds of detection. This reduces Mean Time To Respond (MTTR). Automated playbooks, in fact, could finish important tasks by the time human team members are mobilizing. AI-ChatOps can reduce response time to a fraction compared to conventional workflows—where an event might remain undetected for minutes or hours and need manual coordination. Rapid confinement limits an attacker's stay time, hence lowering possible harm. In sectors like finance, where even a few minutes of unmonitored malicious behavior can have major effects, this speed is especially vital.

ChatOps guarantees developers, operations, and security personnel share the same situational awareness by centralizing all incident data and communications in one channel. Everyone engaged can view the warnings, data, and actions in real time; there are no parallel email threads or distinct siloed discussions. Decisions are taken according on a shared operational picture; this openness removes the "telephone game" of miscommunication. By immediately adding expert knowledge for example, threat intelligence context or suggested actions—into the discussion, AI helps the team to collaboratively make educated decisions. All things considered, this strengthens the feedback loop among team members and promotes a cooperative culture in which incident response is a shared responsibility rather than only the province of a different security team.

AI-driven ChatOps lowers the probability of errors that usually happen when people are tired or hurrying by automating tedious and high-pressure activities. Executing a difficult confinement process at 3 A.M., for instance. Using a tried script is more consistent than manually entering a sequence of instructions in a console. By double-

checking commands—warning if an action seems abnormal or potentially harmful—the AI can also act as a safety net before execution. Playbooks manage the known stages uniformly, therefore reducing the likelihood of forgetting a step or misconfiguring a tool since all activities are carried out via a consistent interface—the chat bot. A single source of truth in the chat log also eliminates mistakes caused by miscommunication; at any given moment, there is no uncertainty regarding what was done and who is doing what.

Every event managed via the ChatOps channel turns into a recorded lecture. Teams can do post-mortems and find what went well or might be better by looking over the chat transcript. This results in future playbook changes and improved methods. From the artificial intelligence point of view, event data—alerts, actions performed, results—can be input into machine learning models. With time, the artificial intelligence becomes better at identifying patterns that really suggest problems as opposed to those that are benign. The model can learn from that input; for example, if the AI identified an event that turned out to be a false positive and the team marked it as such. On the other hand, if the team overrode an AI recommendation in favor of another action that fixed the issue, that knowledge could improve future recommendations. This virtuous cycle indicates that the more the system is utilized, the more efficient it may grow. Reading through the archives helps new team members learn from previous events, hence speeding their ramp-up in knowledge of the security scene of the organization.

AI-driven ChatOps places security monitoring and response straight into the tools and processes previously known to DevOps teams, hence integrating them into CI/CD workflows. Often, IT engineers and developers are already use chat platforms for operations and deployments e.g., running chat commands to deploy code or verify system status. Extending the same interface to security events makes managing those events a natural extension of daily work instead of an outside activity. Since the problems show in a forum they participate in, it motivates developers to be more engaged in security concerns. Furthermore, ChatOps can interact with events in the CI/CD pipeline; for instance, the system can send an alarm in the conversation and even stop the pipeline until fixed if a major vulnerability is discovered in a new build. Real-time enforcement of security gates via chat guarantees that security is really incorporated into the development process. ChatOps with AI, by meeting DevOps teams where they operate in chat helps to fulfill the DevSecOps objective of combining "security as code" and cooperation instead of considering security as an afterthought or distinct silo.

We use a case study in the financial services sector—an industry where high risks and regulatory restrictions make both speed and accuracy of incident response top priorities—to clearly show these advantages. This scenario is a composite based on actual capabilities of AI-SecOps tools and common threat events in a bank:

Imagine a big retail bank with an AI-enabled security monitoring system spread throughout its IT infrastructure. The system comprises a SIEM consuming logs from databases, servers, and network devices; IBM QRadar with Watson or Microsoft Sentinel. Late one night, the AI analytics engine spots an anomaly: a normally idle database server (DBServer12) is unexpectedly sending an extremely high amount of data to an external IP address not seen before. This pattern suggests a potential data exfiltration attempt—a breach in progress.

The artificial intelligence instantly identifies this as a high-priority event. It correlates multiple indications – such as the data amount, the process starting the transfer, and comparisons to known attack patterns – and finds there is a substantial possibility of malicious behavior. The technology creates an incident and triggers a ChatOps alert rather than waiting for a human analyst to see this in a dashboard. Seconds later, a message appears in the bank's "#security-war-room" Slack channel: "Alert: Possible data exfiltration detected on DBServer12". The message offers a summary: e.g., "DBServer12 sent 900MB of data to IP 203.0.113.45 (Russia) at 23:10 UTC, exceeding baseline by 50x." It also offers background information including the username executing the process and a hazard score. The system may say, "This pattern resembles the ACME Corp breach (June 2024) in our knowledge base," since it has historical knowledge. The bot automatically tags the on-call security engineer and pertinent team members (database admin, network engineer, etc.), so their devices get an alert notification right away.

The AI (Watson in this scenario) adds a suggested action with the alert: "Suspected data theft in progress. Based on previous reactions that effectively prevented comparable assaults, I advise turn off database user 'svc_import' and cease DBServer12 outbound communication. The team currently gathers in the Slack channel; some are on laptops, and some are on phones if remote. In the first minute, everyone is staring at the identical bot-posted data. The

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

collaborative conversation starts: "This looks like a probable insider or malware exfiltration; let's contain first," says the security engineer. The database admin adds, "That server holds customer data; we must ensure backups are intact too." Thanks to ChatOps, all these exchanges are happening in one thread visible to all stakeholders (including compliance and comms team members who have joined seeing the high-priority alert).

The team moves forward with the advised containment without wasting time. The security engineer types a command to the bot via the ChatOps interface: @SecBot block-outbound DBServer12. Integrated with the bank's network firewall controller (or SOAR platform), the bot runs a script to isolate that server, hence directly severing its external network access. Another command @SecBot disable-user svc_import runs a script removing the suspicious user's directory service credentials. The possibly harmful data transfer is stopped within one to two minutes of the first notice. Contrast this with a conventional approach without ChatOps/AI: it might have taken the team 15–30 minutes merely to understand what's happening, locate everyone, join a call, and log into several systems to confine the server. Critical containment happened quite instantly here, hence significantly reducing any data loss.

Every action the bot makes is logged in the Slack channel: "Firewall rule added: blocking DBServer12 outgoing traffic" and "User svc_import locked." The IT operations team and compliance officer, also present in the channel, watch these actions. All activities are time-stamped and documented, according to Compliance, which will be quite helpful for reporting. Financial rules sometimes call for a timeline of event activities; the conversation log offers that on the fly—often. The bot returns the outcome showing whether any memory-found malware existed. The communications manager, meanwhile, creates a customer notification form in case this proves to be a breach calling for disclosure. All of this is accomplished via the chat ops channel; no emails are sent.

The immediate danger is nullified within around 15 minutes of discovery. The account utilized was deactivated and the data exfiltration was halted in progress. In a follow-up examination later, it appears out an employee's account was compromised and used to perform an unlawful data export. The harm was little because of the quick reaction. The bank nevertheless follows rigorous breach response policies; generally, institutions have 24 to 72 hours to inform authorities of any major event; in this instance they had a complete report within an hour. The incident ticket stores the record of the Slack incident channel. Using this incident's data, the security team and the AI will retrain detection models in the next days, noting the indicators (so that if a similar pattern arises, it might be identified even sooner).

This situation draws attention to some practical benefits of AI-driven ChatOps. A significant data breach was avoided by the quickness of identification and response. The cooperation and openness guaranteed that everyone required was engaged and knowledgeable, and nothing was missed or postponed depending on one individual. Given the AI had performed some of the analysis that people would typically take longer to complete, the AI recommendations helped the team feel empowered to move fast. Moreover, the whole event was recorded without difficulty, which is very important for finance compliance and post-incident assessment. It closely resembles a real case study in which a bank suffering from alert fatigue and slow, manual replies enhanced their results by using automation and integrated tools In that genuine scenario (NKGSB Bank), previous to automation the security staff was overloaded and responses were sometimes too delayed, however after adopting an integrated solution, they considerably decreased alert volumes and reaction times. Our situation shows the same thesis: by using AI and ChatOps integration, financial institutions (and companies in general) can remain one step ahead of risks while keeping the audit trails and assurance that regulators and stakeholders need.

V. DISCUSSION

Although the outcomes of AI-driven ChatOps for incident response are encouraging, there remain significant issues and points to address. Using this method is not without challenges, and some restrictions have to be accepted. We also investigate future paths that can improve even more the efficacy of AI-ChatOps in DevSecOps.

A. Issues and Constraints: Integration complexity is one significant difficulty. Getting all these to smoothly interact with a chat platform calls significant engineering effort as organizations usually have a varied mix of security products (SIEM, EDR, firewalls, ticketing systems, etc.). Some tools lack developed APIs or chat connections. Teams might have to leverage middleware such as a SOAR platform or create bespoke scripts to close any gaps. Particularly in companies with older systems, this integration effort may be substantial. ChatOps also brings a learning curve and

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

cultural change. Team members must adjust to executing vital tasks in a chat interface. Some might first doubt or feel uneasy about a bot altering systems. Building confidence in the automation is vital; this usually means thorough playbook testing and maybe beginning with read-only or suggest-mode bots before letting them modify systems. Over-reliance on artificial intelligence also poses a danger. An error in the AI's reasoning could result in wrong behaviors if team members start to blindly follow AI advice. For instance, a false positive warning can cause a pointless server shutdown. Human validation the human-in-the-loop concept should stay in place for significant activities to help to offset this. Strict access constraints should stop the AI from acting outside specified limits.

Another constraint is that although advanced, existing artificial intelligence models are not flawless. Attackers could find methods to avoid machine learning detection or perhaps control artificial intelligence using adversarial inputs or by setting off a high number of noise alarms to divert the AI. The system must be built to handle false positives and false negatives gracefully. Should the team grow overly reliant on the AI to generate alerts, false negatives—missed events—are cause for worry. Therefore, as a backup, conventional monitoring and redundant detection techniques should remain in effect. In the chat, false positives can create "alert spam," which might make significant alarms go unnoticed (the traditional boy-who-cried-wolf issue). Tuning the AI models and alert thresholds is an ongoing effort; the AI-ChatOps system must contain feedback loops as discussed to continuously improve accuracy. Furthermore, scalability and performance must be considered: a big corporation could produce daily thousands of notifications; sending these into a chat channel would flood people. Allowing the AI to filter and summarize solves the problem, however that calls for extremely strong artificial intelligence performance once more.

Another important topic of conversation is the security of the ChatOps channel itself. Ironically, the chat platform might be a conduit for attackers if properly safeguarded. A foe with access to the incident channel or the bot's credentials could view private incident data or perhaps send harmful orders. Strong authentication multi-factor for human users, and secure tokens for bots—encryption, and access controls are thus required. Like a privileged account, the bot should be treated; its actions should be checked and its credentials to start automation kept safe. There should also be fail-safes: for example, some harmful activities would need dual clearance via chat (two authorized users verifying) to carry out, therefore preventing a situation where either AI error or a single compromised person could inflict damage.

From a people viewpoint, there is a change management element. To make the most of the latest technologies, teams require training. Some conventional incident responders might resist the chat-centric strategy and prefer working in known SIEM consoles or over phone bridges. Showing the advantages and offering incremental adoption perhaps beginning ChatOps for less important events and growing can help. Defining procedures on how the conversation will be used etiquette, responsibilities, how to escalate if required outside chat, etc. is also crucial. Ensuring that the ChatOps method supports business or regulatory policies e.g., data retention, privacy is essential; chat logs could include sensitive information that must be handled per policy.

B. Future Developments: The path of technology indicates that many of these problems can be lessened despite these difficulties and the possibilities will continue to get better. The employment of sophisticated artificial intelligence, particularly generative models, in the ChatOps process is one significant area for future development. We are already witnessing the start of this: large language models (LLMs) like GPT-4 may be used to analyze complicated queries and produce human-like explanations. A future ChatOps bot in incident response could use an LLM to analyze unstructured logs or suggest replies more contextually aware. For instance, instead of pre-programmed replies, the bot could talk to analysts: a straightforward question from an analyst could be, "What is the possible impact of this incident?" and an AI like GPT could combine an answer using incident data and existing knowledge. An AI like GPT may combine an answer from incident data and known knowledge with an analyst's plain language question, "What is the possible impact of this incident?" A step in this regard is Microsoft's announcement of Security Copilot, an LLM-based AI assistant for cybersecurity meant to help analysts by summarizing events and making recommendations for questions that might readily connect with a ChatOps interface in the future. Similarly, Atlassian's recent AI features that summarize chat talks into Jira tickets hint at a future when routine documentation is completely automated by interpreting the natural language conversation.

Wider integration across company borders is another anticipated development. ChatOps is now mostly confined to one team or corporate setting. We might eventually have safe cross-organization ChatOps to share threat intelligence or

© 2025 IJMRSET | Volume 8, Issue 2, February 2025| DOI:10.15680/IJMRSET.2025.0802085

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

coordinate responses to large-scale events, such as vital vulnerabilities impacting multiple businesses. AI might mediate these communications, drawing in pertinent information from outside sources in real time. Improvements in standardization as well—such as the creation of shared formats for incident data and response playbooks—could help various tools to work together in an AI-ChatOps system. Examples include the STIX/TAXII standards for threat intelligence and the developing OASIS CACAO standard for cybersecurity playbooks. This implies that a company may install a new detection tool and have it begin feeding the chat bot without significant custom programming.

Techniques for human-AI cooperation will also develop. Future systems could be conversational instead of the AI simply generating alarms and the people reacting. For example, the AI may pose the team clarifying questions to make the interaction more engaging: "This looks like a possible false positive. Do you want me to ignore similar alerts for the next hour?" As the AI may assume more decision-making under direction, this idea of conversational artificial intelligence for security could significantly improve productivity. User interfaces of chat platforms are expected by us to change to better fit such workflows—for instance, richer message formats with integrated dashboards or mini-forms for approval (some of which already exist in platforms like Slack and Teams but could be more AI-driven).

Research on explainable artificial intelligence (XAI) for security will be crucial in terms of handling present constraints. Human operators will trust and understand the AI more if it can explain why, it made a recommendation—for example, by stressing the elements that caused it to flag an incident high priority. They will also be able to rectify it if required. This may be incorporated into the chat conversation; the bot might offer a "explanation" upon request for any alert it posts.

From an organizational process perspective, when AI-driven ChatOps demonstrates its worth, we may eventually witness incident response processes being restructured to officially incorporate AI and chat integration. Through cooperative technologies, cybersecurity frameworks and laws might change to recognize controls and record-keeping. Businesses might begin to track MTTR and other measures particularly on how quickly the ChatOps loop addressed an event and utilize that to propel changes.

C Successful AI-driven ChatOps has implications outside only IT incident response. This method shows how artificial intelligence may be integrated into team processes to enhance human potential. Lessons acquired here could apply to other fields such business continuity planning, IT operations (for outage management), customer support (where bots support engineers in conversations), or perhaps others. In every situation, the formula of detect cooperate – act may produce quicker and more dependable results using artificial intelligence accelerating processes and offering insights.

Still, thoughtful thought of the human element is vital. The aim is to enhance, not supplant, human decision-making. Our conversation emphasizes that although automation, human supervision is essential to manage new or complicated circumstances that AI may not completely understand. Incidents that defy automation will always exist multi-faceted attacks needing inventive investigation, zero-day attacks with no known pattern. Even if the AI retreats, the ChatOps framework still offers value as a coordinating tool for such people.

All things considered, although issues like integration effort, trust, and accuracy must be controlled, the future of incident response seems to be moving toward more usage of artificial intelligence inside cooperative settings. Much way CI/CD pipelines and infrastructure-as-code are today, AI-driven ChatOps could become a normal component of DevSecOps toolchains as the technology and techniques evolve. Early adopters will benefit from resilience through speed and shared intelligence; those clinging to segregated and manual procedures may find it more difficult to keep up with the speed of both cyber threats and software delivery.

VI. CONCLUSION

In a DevSecOps environment, AI-driven ChatOps is a major development in how companies manage security events. In this paper, we reconstructed the concept of combining artificial intelligence with ChatOps and shown that it brings together the best of two worlds: the speed and analytical capacity of computers with the collaborative and creative problem-solving talents of humans. DevSecOps teams may identify events earlier, respond more quickly, and cooperate more efficiently by incorporating artificial intelligence features into chat-based processes—all without

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

departing their daily communication channel. From a scattered, sometimes frantic rush, the incident response process becomes a clear, consistent approach in line with continuous delivery norms.

From NIST's demands for quick response to industry case studies that underscored failures of solely manual processes to triumphs with automated, collaborative solutions, our review of existing technologies and literature revealed considerable support for this integrated approach. We outlined a process by which artificial intelligence tracks systems and delivers detailed alerts to a ChatOps channel, where the team and AI work together to triage and contain the threat. Among the outcomes are significantly faster reaction times, better development and operations team communication, and incident information collection for ongoing enhancement. The financial industry example showed how AI and ChatOps working together may prevent a possibly catastrophic breach in minutes, a scenario increasingly within reach given capabilities like IBM QRadar, Microsoft Sentinel, and Palo Alto XSOAR under discussion.

The difficulties in deploying AI-driven ChatOps, including integration issues and the necessity to preserve human oversight, received critical attention. Though genuine, these difficulties are conquerable with thoughtful design, training, and system iterative tweaking. Moreover, in settings where every second matters during an event, the advantages surpass them. We also saw that this strategy promotes a culture where security is "baked in" engineers and analysts work together in one forum, hence sharing responsibility instead of isolating function. This culture change is consistent with DevSecOps ideas and may result in better security-aware development methods generally.

Looking forward, we expect that AI-driven ChatOps will be more prevalent and even required. Cyber-attacks remain to become more complex; manual reactions can no longer maintain pace. On the other hand, software delivery is quickening, which drives companies to seek security measures that don't hinder creativity. AI-driven ChatOps increases security responsiveness without removing engineers from their flow, therefore addressing both demands. Future developments in artificial intelligence, particularly in natural language interpretation and knowledge representation, hold the potential to make these chat assistants even more smart and useful. One may imagine that shortly a security alert will be more a staged discussion than a cause for alarm; an AI assistant has already completed the normal duties and offers the team obvious resolution choices.

Ultimately, combining artificial intelligence with ChatOps is a strong technique to automate the monotony and delay in incident response while maintaining human control of important decisions. Companies that adopt this strategy can preserve the fast speed of DevOps delivery, reduce the effect of events, and attain a better security posture. So far, the studies and case data point to AI-driven ChatOps as a revolutionary technique for DevSecOps rather than only an incremental improvement. Such cooperative automation will probably be a cornerstone of how we keep secure and resilient systems in an age of continual change as threats alter and the instruments at our disposal enhance.

REFERENCES

[1] NIST, Computer Security Incident Handling Guide (SP 800-61 Rev.2), National Institute of Standards and Technology, 2012.
[2] S. Regan, “ChatOps is a collaboration model that connects people, tools, process, and automation...,” Atlassian (Incident Management blog), 2020.
[3] Palo Alto Networks, **“Faster incident response and reduced alert fatigue at NKGSB Bank,”** Case Study, Palo Alto Networks, 2023.
[4] Microsoft, **“Use a Microsoft Sentinel playbook to stop potentially compromised users,”** Microsoft Learn (Tutorial Documentation), 2023.
[5] IBM Security, **“IBM QRadar Advisor with Watson,”** IBM Product Brief, 2017.
[6] J. Goh, **“Frequently Asked Questions About Security ChatOps,”** Palo Alto Networks Blog, Nov. 2017.
[7] S. Brooks, S. Martin, and M. Song, **“AI-Based ChatOps: Enhancing Collaboration and Incident Response in DevOps Teams,”** Research Article, Jan. 2025.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com