Team 1- COMP9447- Indicators of Compromise & (Automated) IR Playbook

Table of Contents

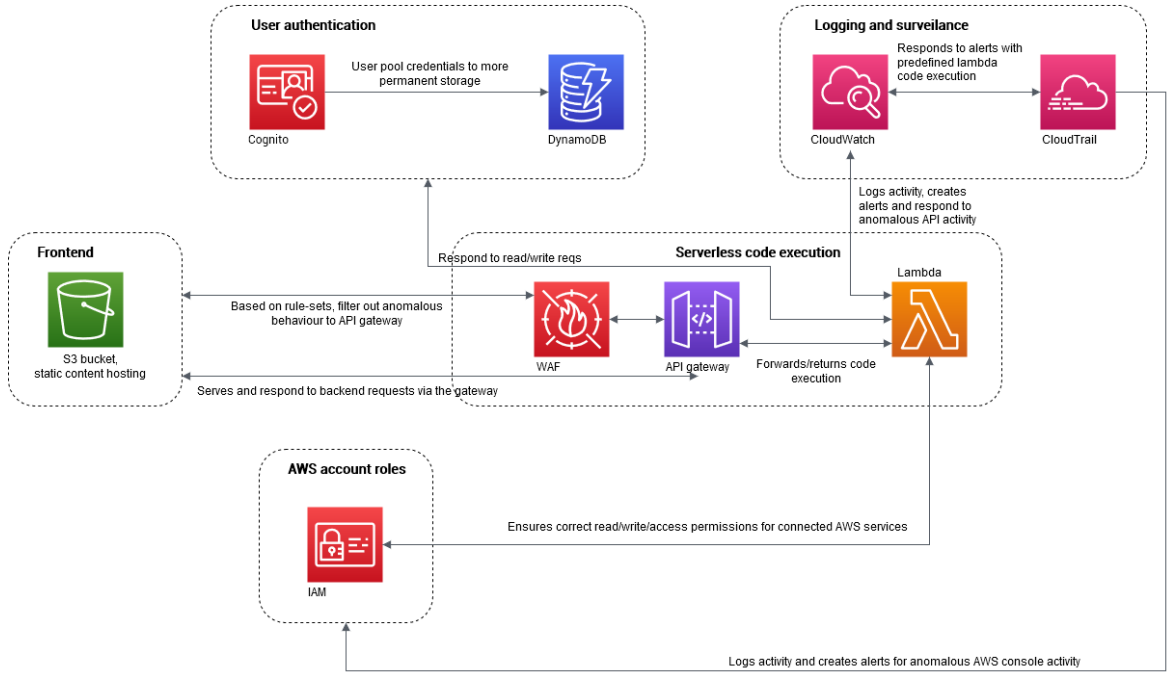
Introduction

This document contains both high-level descriptors of most threats to a Medical CRM and descriptions of fine granularity about incident response to a selection of threats. Initially, threats are defined by the STRIDE threat model and then a few are selected to be refined further. The responses to selected security incident use cases are completely automated by the extensive use of AWS lambda and the workflows for those are documented in the playbook diagrams.

Typically, incident response to customer domain AWS breaches are manual and reactionary. This project attempts to reduce the latency between detecting a threat, acting against the threat and preventing security issues in a customer configured AWS environment. Furthermore, these security structures are continuous, reducing the need for manpower either dispersed globally or outside of typical business hours.

After examining this paper, you will understand how you can incorporate AWS serverless computing into your AWS stack to further improve the security of your AWS architectures. Through the adoption of these strategies, you can build AWS stacks that safeguard your data, running services through appropriate access control, and automatic response to security incidents.

The Architecture of a Medical CRM



The above diagram shows a fully cloud based basic CRM (Customer Relationship Management) solution. This solution is broken down into several basic functioning features:

- Authentication
- Data retention
- CRUD operations
- Front end
- Continuous logging and surveillance

Medical CRMs has additional privacy requirements (usually stated by local laws) on top of a regular CRM. Greater emphasis on data security is needed, such as security of data in transit and at rest, enabling traceability on actions and keeping people away from data.

Threat Model Against a Medical CRM

Threat Vectors

- Active
    - Via the frontend
        - XSS
        - Brute-force login field
        - Invalid user input
            - Causing information leakage from return values
        - DDoS attack (outsider)
    - Via AWS
        - Privilege escalating via adding IAM policy changes (insider)
        - IAM root account / demise of service (insider)
        - API endpoints
            - Misconfigured security protocols
        - Incorrect authorisation settings on API endpoints
- Passive
    - Traffic analysis via Wireshark (examining json and HTTP responses)
    - Spoofed emails that impersonate the CRM's sent emails (outsider)
    - Spoofed CRM login website (outsider)

Table View of Threats

Threat model based on STRIDE threat model framework (spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege)

#	Elements	Interactions	S	T	R	I	D	E
1.1	DynamoDB	Malicious CRUD operation on DynamoDB entries		x		x		
1.2		Massive reads from database to remote				x	x	
2.1	Cognito	Invalid/spoofed JWT token sent for authentication	x		x			x
2.2		Brute-force password attempts (on CRM frontend)					x	x
2.3		Malicious login into CRM account	x		x	x		
3.1	IAM	Root account with faulty IAM role settings	x	x	x	x	x	x
3.2		Compromised password to any IAM user	x	x				x
4.1	CloudTrail	CloudTrail instance turned off via AWS Console		x	x			
4.2		CloudTrail logs tampered (via deletion)	x	x	x			
5.1	Frontend/S3	Cross site scripting into S3 bucket or user's browser	x			x		x
5.2		Exposure of auth token			x	x		x

5.3		Invalid user inputted data		x		x		
5.4		Buckets permission is public access (allow all)	x	x	x	x		x
5.5		Cross site request forgery	x		x	x		
6.1	WAF	Under/misconfiguration of rulesets		x		x		x
6.2		Slowly transmitted data being sent					x	
7.1	Lambda	Incorrect permissions	x	x		x		x
7.2		Event injection	x	x		x		x
7.3		Exposed parameters				x		

Implications of Threats

1. **DynamoDB:**
    a. Leak personal medical information or destroy the integrity of patient data by performing illegal alterations
    b. Can cause a denial of service attack causing the critical CRM to cease operations or obtain medical records without authorisation
2. **Cognito services**
    a. **JWT tokens**: Tokens are rotated every 5 minutes on the Cognito service; however, this service is used to authorise APIs requests. If the token was successfully forged, then the attacker has authorisation over API requests (such as CRUD operations)
    b. **Brute force attack / leaked password**: If the attacker is successful, then they have access to sensitive medical data. In addition, the mass requests would lock the system up if it did not employ rate-limiting.
    c. **Incorrect authorisation:** Attacker gains sensitive medical data
3. **IAM**
    a. Access to **root account:** An attacker with access to the root IAM account can create/tamper/shutdown AWS services and exposure to personal and payment details submitted to AWS.
    b. Access to **less privileged accounts**: An attacker with a non-root IAM account is still able to map the services used and if the role permissions are set incorrectly, it provides a pathway to privilege escalate.
4. **CloudTrail**
    a. **Instance shutdown:** This allows the attacker to conduct malicious AWS activity without being logged
5. **Frontend and S3 buckets**
    a. **XSS**: Allows the attacker to inject malicious code, which can be either stored in the S3 bucket or reflected. This can cause the user's session cookie or Cognito authorisation token.
    b. **Exposure of (Cognito) auth token:** The attacker can hijack the user's (CRM) account or used to send malicious API calls using the leaked authorisation token.

c. **Invalid user inputted data:** While DynamoDB is NoSQL, it is not immune to injection attacks. The attacker can use the scan() function and comparison operators to form an injection payload. This in turn can leak (potentially sensitive) fields stored in the database.
d. **Incorrect bucket permissions**: Shows the structure of the website and possibly AWS architecture (such as other buckets that are being maintained by the same user). Also possibly allows for the modification of S3 bucket objects
e. **CSRF:** While requiring the attacker to spoof the medical CRM, this attack can cause unwanted action (such as forging a prescription, creating a privileged new user, or exfiltrating data)

6. **WAF**
    a. **Bad Rulesets:** Incorrectly configured WAFs allow XSS, DDoS and other web application attacks to occur
    b. **"Low and Slow" Attack:** Generally mitigated with Amazon's Elastic Load Balancing, however if undetected by load balancing or the WAF causes a slow increase in resource usage. This causes requests to be served slowly and increases AWS billing.
7. **Lambdas**
    a. **Incorrect Permissions:** Increases the attack surfaces, allows attack of the AWS account and running services
    b. **Event Injection:** Allows arbitrary code execution
    c. **Exposed parameters:** This allows the attacker an insight to the AWS service (e.g. the direct S3 bucket). EC2 service may also expose the environment variables (containing the STS variables and/or S3 bucket info)

Threats Covered in This Project

The threats that are in scope of this project are as follows:

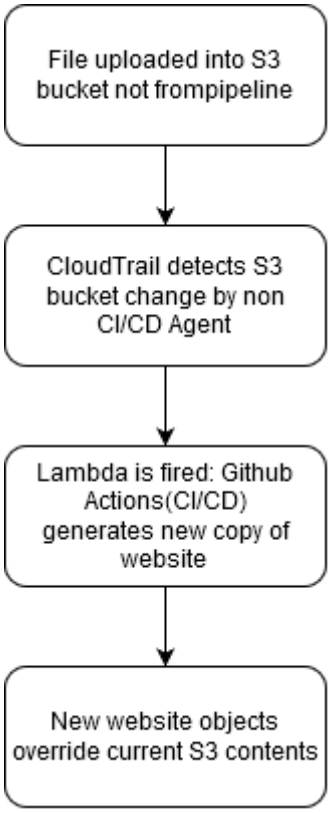
- 1.1-2
- 2.2-3
- 3.1
- 4.1
- 6.4

Internal/External Incident: Attempted Vandalism

Indicators of Compromise

1. New files are added to the S3 bucket not via the CI/CD (GitHub Actions)
2. CloudTrail logs show an edit to any of the files' contents in the S3 bucket

Incident Response Runbook: Extraneous file added to S3 bucket

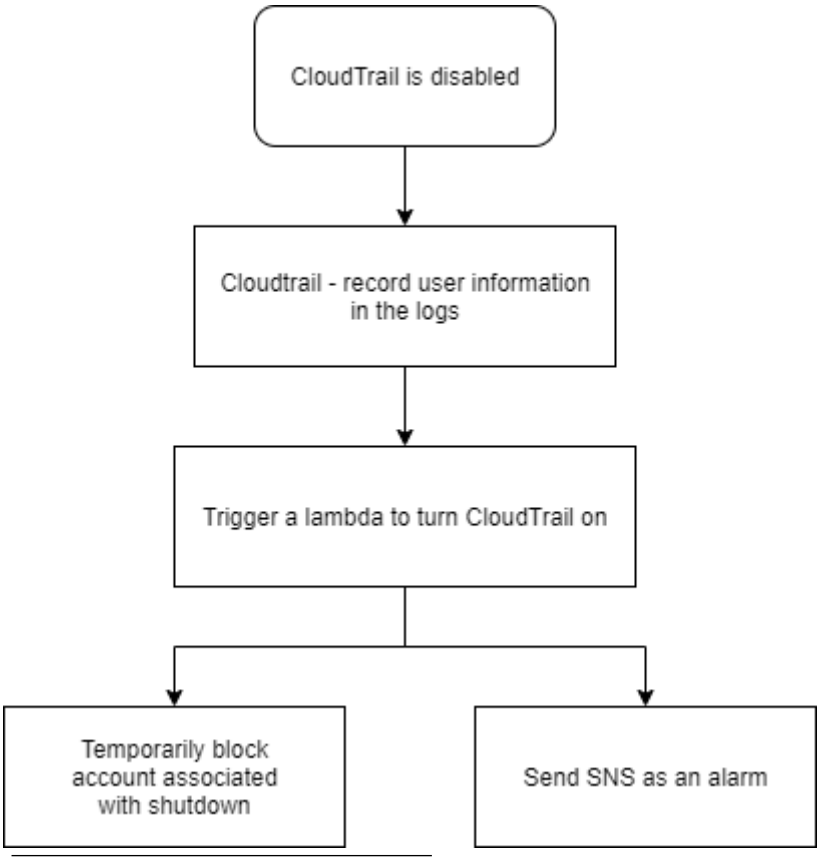


Internal Incident: Compromised AWS Account

Indicators of Compromise

1. Typical attacker behaviour as follows: unusual/excessive database reading/writing, altering billing account details, tampering with logs
2. CloudTrail instance is tampered with
3. IAM with more privileged roles are being assigned

Incident Response Runbook: CloudTrail Disable Use Case

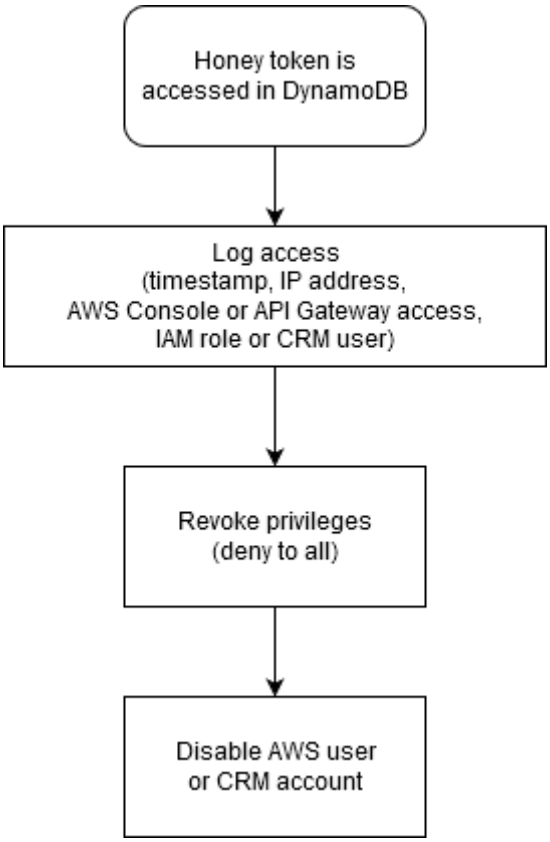


Internal Incident: Data Exfiltration

Indicators of Compromise

1. Enormous amount of read volume
2. Unexpected log (e.g. data transfer to an unknown endpoint)
3. Increased AWS billing activity deviating from normal usage (indicating increased usage)
4. Honeypot account record is accessed

Incident Response Runbook: Honey Token Use Case

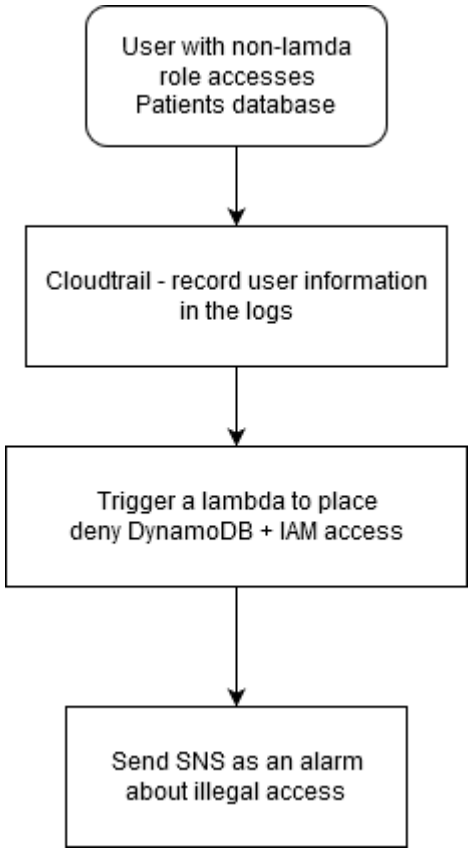


Internal Incident: Non-Patient Reading/Writing Patient Data

Indicators of Compromise

1. Logs indicating a non-patient attempted to access the Patients Table
2. Logs indicating that the attacker attempted to remove deny all role from administrator account

Incident Response Runbook

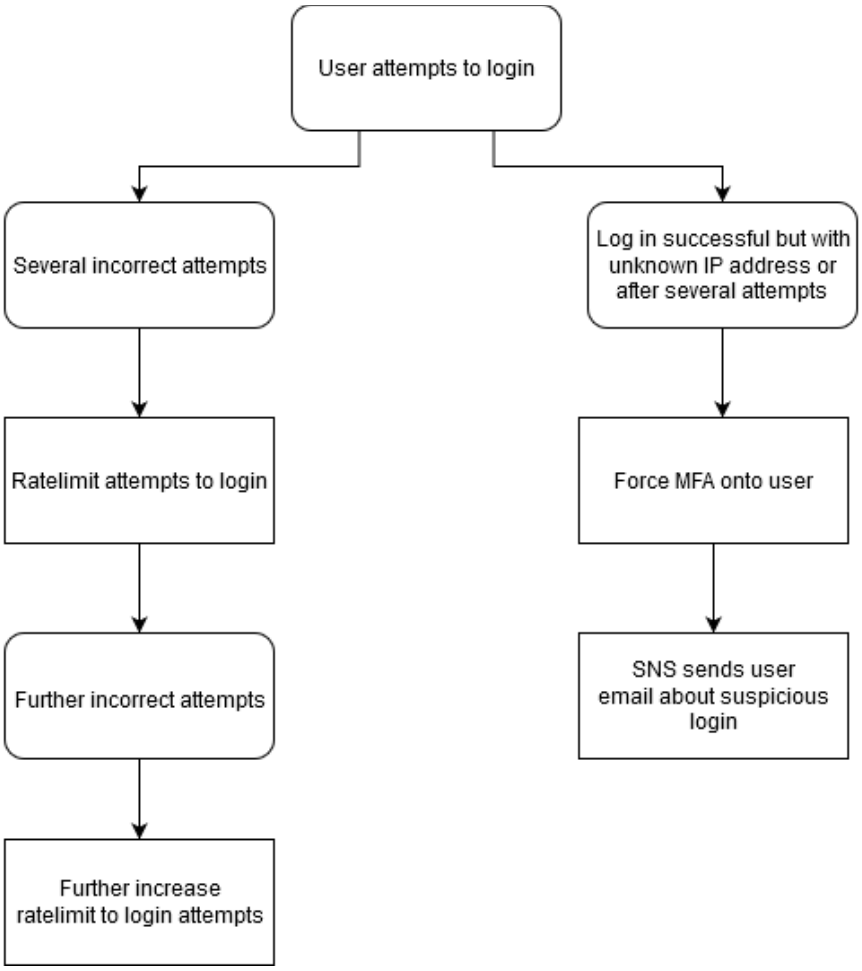


External Incident: Compromised CRM Account/Brute-force Attack

Indicators of Compromise

1. Failed logins using user accounts that does not exist
2. Access from unknown IP/device/IP location
3. Log-in attempts on one account within a short period of time from different IPs around the world
4. CloudTrail logs indicating increased Cognito usage that differ from normal amount of traffic
5. Typical attacker behaviour as follows: unusual/excessive database reading/writing, altering medical details

Incident Response Runbook: Attempted password exfiltration



External Incident: API Attack

Indicators of Compromise

1. Elevated request levels
2. A lot of requests coming from certain IP addresses
3. Malicious request signature detected (SQLi, XSS)
4. Request matches IP reputation list

Incident Response Runbook