# Political Implications of Zero Trust Cybersecurity Policy

Jaden Walker

CYSE-425W

Dr. Shideh Yavary Mehr

6/18/2025

The BetMGM Data Breach stands out as one of the most revealing examples of what happens when a fast-growing digital platform does not invest enough in basic security architecture. In late 2022, the company suffered a breach that exposed millions of customer accounts, revealing deep vulnerabilities in how online gambling systems are built and managed. What made this incident important was not just the number of people affected but how simple the attack methods were. The breach showed that even major companies with large user bases can still overlook the fundamentals of authentication, API protection, and internal system monitoring when performance and revenue take priority over long-term security planning. Much like how Zero Trust became necessary after repeated failures, the BetMGM incident highlighted how reactive cybersecurity approaches create larger problems down the road.

The architecture behind BetMGM's platform played a big role in the breach. The system relied on cloud services spread across multiple AWS regions, with PostgreSQL handling transactions and MongoDB storing session data. While this setup allowed the platform to scale quickly, it also created uneven security controls. The company continued relying on simple username-and-password logins for the main customer accounts. This choice introduced unnecessary risk because attackers increasingly use credential stuffing campaigns to exploit reused passwords. In my opinion, this reflects a pattern we see in many fast-paced tech environments where convenience wins out over secure design. When developers focus heavily on speed and performance, security layers become optional instead of essential.

The attackers used residential proxies to disguise themselves and spread login attempts across thousands of IP addresses. Because BetMGM only rate-limited logins per IP instead of per account, these attempts blended in with normal traffic. Once inside an account, the attackers quickly realized that BetMGM's session tokens were weak. The JSON Web Tokens used by the

platform were predictable and not properly validated, meaning the API gateway trusted any token that looked correctly formatted. This allowed attackers to forge tokens that gave them access to accounts they never legitimately logged into. This flaw alone shows how dangerous it is when authentication systems are not configured to verify token signatures or enforce expiration rules.

The attackers continued digging through the system and uncovered more architectural gaps. One of the clearest weaknesses was an insecure direct object reference in the account management API. The threat actor needed to change the user ID parameter in a request in order to retrieve information belonging to another customer. The system only checked whether a session token existed; it did not check whether that token matched the user whose data was being accessed. This is the type of mistake that secure development practices are meant to prevent, but it often happens when teams lack strong security oversight. MongoDB queries also had issues. Instead of using parameterized queries, some MongoDB operations were built through string concatenation, which made them vulnerable to injection attacks. These problems show that the breach was not the result of one major flaw but a chain of small but impactful oversights.

The monitoring systems should have caught the initial attack, then alerted the SOC for immediate remediation, but the stack was not configured correctly. BetMGM relied on the ELK stack for log aggregation, but they did not integrate some sort of “false positive detection” system. This has become a widespread problem across multiple SOCs because it desensitizes the SOC analysts and engineers. The way to prevent this would be integrating some automation to alleviate most of the false positives to ensure each alert is treated, instead of ignored. Cloudflare protected the network from large DDoS attacks, but the platform did not have effective application-level rate limiting. As a result, the attackers’ requests quietly passed through.

Additionally, API error messages revealed too much internal information, including stack traces and database schema details. In cybersecurity, these kinds of leaks give attackers a roadmap for how the system works.

Several major encryption and communication issues also worsened the impact of the breach. While BetMGM encrypted payment card data using strong methods, other sensitive data, such as Social Security numbers and dates of birth, were stored using reversible encoding. Passwords were hashed with bcrypt, but the cost factor was too low to resist modern GPU-based cracking. Even worse, internal microservices communicated using plain HTTP instead of HTTPS, operating under the outdated idea that the internal network was safe by default. Once attackers gained access to one service, they could intercept traffic between services, collect session tokens, and listen in on sensitive data being transferred. This reflects a common but flawed assumption in system design: that internal systems do not need to be protected as strongly as external ones.

The data exfiltration process was slow and intentional. Instead of taking massive amounts of data at once, the attackers extracted information in small pieces over a long period. Stolen data was temporarily placed into misconfigured S3 buckets that had public read permissions. Before the data was moved, it was compressed and encrypted, making it harder for detection tools to recognize what it contained. The attackers also used techniques to make the traffic seem as though it was going to a common destination. All of this happened partly because BetMGM did not have a strong data classification strategy. Sensitive data was not segmented, field-level encryption was not used, and backups lacked server-side encryption. The other part of the attack surface was exposed because MFA was not required for any admin accounts, and many systems have started requiring MFA for all end users as well.

The company's incident response approach revealed operational issues. The security team was slow to react to the incident because there was no clear, tested SOP for events like this. At first, they reset user passwords but did not invalidate active sessions, meaning attackers stayed logged in. BetMGM also kept application logs for only seven days. This short retention period made it nearly impossible for investigators to determine the full timeline of the breach. This is a reminder that incident response is not just about reacting to the attack but also ensuring that the organization has the tools and policies needed for proper analysis afterward.

After the breach, BetMGM had to make major architectural changes. The company paid higher than top dollar for experienced engineers and moved toward a Zero Trust-style model by replacing its flat internal network with microsegmented zones. All communication between services was encrypted with mutual TLS, and a service mesh was introduced to handle authentication and authorization between microservices. The authentication system was rebuilt using WebAuthn, which supports passwordless login through biometrics and hardware security keys. Session management became stricter, with short-lived tokens and device binding to prevent token replay attacks. API security also improved through OAuth 2.0 with PKCE, standardized API responses, and multi-layer rate limiting. BetMGM implemented a defense-in-depth approach by adding web application firewalls, runtime application self-protection, and database activity monitoring.

Across the industry, the BetMGM breach exposed many shared weaknesses. A large number of gambling platforms used outdated frameworks, weak token generation, and inconsistent security practices. This incident pushed organizations to adopt more formal security frameworks like the NIST Cybersecurity Framework, ISO 27001, and the OWASP Top 10. Many companies also strengthened their CI/CD pipelines by adding static code analysis, dynamic testing, and

dependency scanning. Third-party risk management became more important because some vulnerabilities came from external components. Companies began requiring stronger security checks from vendors and used software composition analysis to track open-source libraries.

In conclusion, the BetMGM incident shows the consequences of growth without strong security planning. The attack succeeded because of small design oversights, weak authentication controls, and inconsistent encryption. As online platforms handle more money and more personal information, security cannot be something added later—it must be part of the system from the beginning. Similar to the push for Zero Trust across the government, the lessons from this breach show that long-term protection requires structural planning, strong leadership, and consistent investment. The BetMGM breach serves as a reminder that attackers only need a few gaps to succeed, and it is the organization's responsibility to make sure those gaps do not exist in the first place.

- Peters, G. (2021). Strengthening Federal Cybersecurity. Senate Homeland Security and Governmental Affairs Committee. https://www.hsgac.senate.gov/
- Rose, S. W., Borchert, O., Mitchell, S., & Connelly, S. (2021). Zero Trust Architecture. National Institute of Standards and Technology (NIST). https://www.nist.gov/publications/zero-trust-architecture
- Implementing a Zero trust architecture | nccoe. (n.d.-b). https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture