



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Assessment 01

Title:

Research and case scenario in Data Security and Compliance.

Applied Computing.
Data Security and Compliance.
ACCA5034.

Dr. Ayman El Hajjar (Lecturer)

University of Wales Trinity Saint David

(London Campus)

Student Name: Mr. Shugirthan Rasaratnam Amirthalingam

Student Number: 2430884

Group: 01

Team: 01

Date: 10/08/2026

Table of Contents

Task 1 : Research in Data Security and Compliance.....	3
1.1 Introduction	3
1.2 Core Principles of Zero Trust: Anticipate and Protect	3
1.3 Continuous Detection and Automated Response	4
1.4 Implementation Challenges.....	5
1.5 Efficacy Compared to Traditional Models.....	6
Conclusion (Task 1).....	7
References (Task 1).....	8
Task 2 : Case Scenario: Designing a Cybersecurity Programme based on Equifax	9
2.1 Introduction Context.....	9
2.2 Organisational Structure and Accountability	9
2.3 Policies and Procedures	11
2.4 Governance and Strategic Alignment (NIST CSF 2.0).....	12
2.5 Risk Management and Assessment.....	14
2.6 Compliance, Incident Response, and Protection Controls.....	15
Conclusion (Task 2).....	16
References (Task 2).....	17
Appendix.....	18
Appendix A: Equifax Attack Vector Diagram (Conceptual).....	18
Appendix B: Zero Trust Mitigation Mapping	18

Task 1 : Research in Data Security and Compliance

The Zero Trust Architecture: A critical analysis of its principles, implementation challenges, and efficacy compared to traditional models.

1.1 Introduction

For much of the last two decades, enterprise security has followed a "castle and moat" logic: build a strong perimeter and treat anything inside it as trustworthy by default. That assumption has not aged well. Cloud adoption, a workforce that increasingly connects from outside any single network boundary, and attackers patient enough to sit inside a compromised environment for months have all eroded the idea that "inside the firewall" is a meaningful security boundary [1]. The shift accelerated sharply after 2020, when large-scale remote work forced organisations to expose internal applications to the open internet almost overnight, often through VPN concentrators that were never designed to carry that volume of traffic or that level of scrutiny. Software supply-chain incidents such as the 2020 SolarWinds compromise added a further complication: a trusted vendor's own infrastructure became the attack vector, which meant that "trusted" could no longer be treated as a fixed property of a device, a network segment, or even a well-known supplier.

Once a perimeter is breached, lateral movement across a flat internal network tends to be straightforward, which is precisely what has allowed several of the more damaging breaches of the last decade to escalate from a single foothold into full-scale data loss. Zero Trust Architecture (ZTA) responds to this not by building a bigger wall, but by removing the notion of a trusted interior altogether. The term is generally credited to John Kindervag, writing at Forrester Research, and was later formalised by the National Institute of Standards and Technology in SP 800-207 around a single guiding principle: "never trust, always verify" [2]. This section examines what that principle requires in practice how organisations anticipate, protect against, detect, and automatically respond to threats under a Zero Trust model before turning, in the second half of this report, to how those same principles translate into a concrete cybersecurity programme.

1.2 Core Principles of Zero Trust: Anticipate and Protect

Rather than concentrating controls at the network edge, ZTA pushes verification down to the level of the individual resource, application, and dataset. Anticipating threats depends heavily on visibility: an organisation that does not know what assets it holds, how data moves between them, or how users normally behave has little basis for predicting where an attacker will strike. Zero Trust literature frames this as mapping the "protect surface" the specific data, applications, assets, and services that actually matter, such as a single database of customer records, rather than an entire, sprawling attack surface that spans every device and application an organisation owns [3]. Because the protect surface is orders of magnitude smaller than the attack surface, it becomes realistic to build controls tight enough to notice when something unusual happens near it, which is simply not achievable when the goal is to defend everything equally.

Protecting that surface then rests on a small number of principles that recur across most implementations. Access requests are meant to be explicitly verified using whatever signals are available identity, location, device posture, and the sensitivity of the data being requested rather than granted on the basis of network location alone. This is a meaningful departure from older role-based access control models, where a static role, once assigned, tended to remain valid regardless of context; a Zero Trust policy engine instead evaluates context continuously, checking, for instance, whether a device carries a current EDR agent, whether its operating system is fully patched, and whether the login pattern is consistent with the user's history before a request is even considered. Privilege is kept as narrow as possible, typically through just-in-time and just-enough-access policies that expire once the immediate task is complete [4], which avoids the common failure mode where an employee accumulates access rights across several roles over a career and never has any of them revoked. The network itself is broken into small, tightly controlled segments, so a successful compromise does not translate automatically into free movement across the estate. In practice this is enforced through a Policy Decision Point that evaluates each request against identity, device health, and contextual risk, and a Policy Enforcement Point that sits in line with the traffic and carries out its decision (Figure 1), meaning access is never simply granted once and left standing.

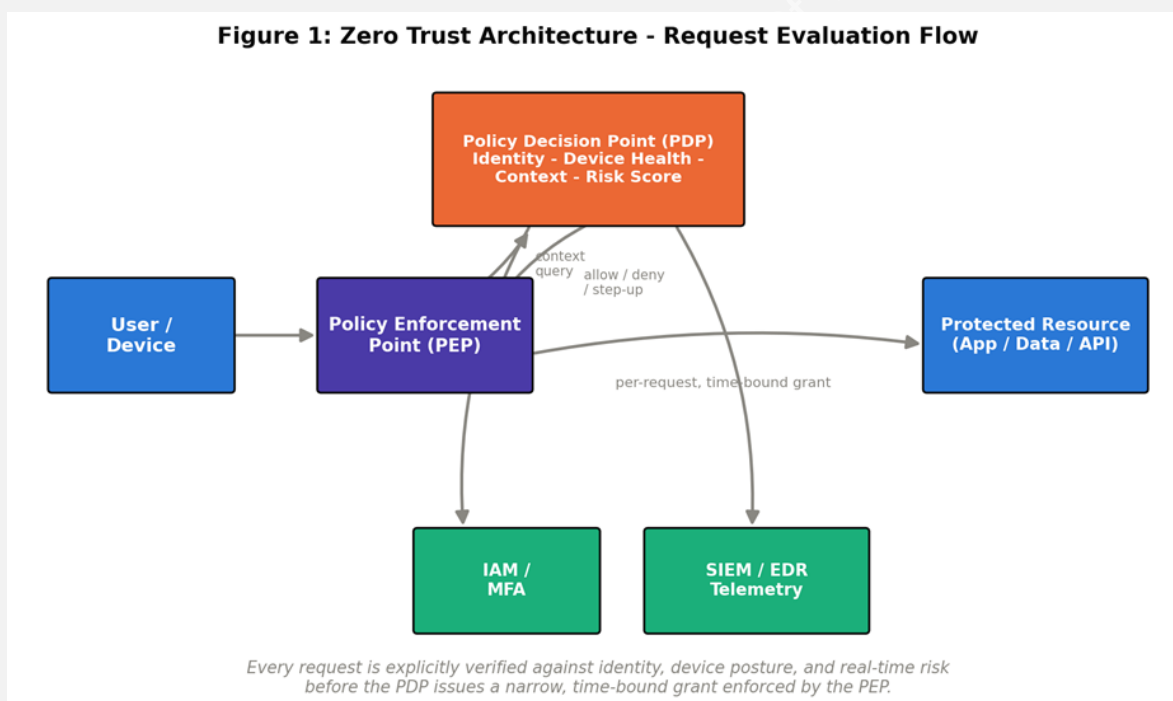


Figure 1: Zero Trust request evaluation flow the PDP/PEP decision loop.

1.3 Continuous Detection and Automated Response

A further departure from traditional models is that authentication is not treated as a single event at login. ZTA assumes conditions change continuously, so detection has to be continuous rather than periodic. This is usually achieved by tying Identity and Access Management systems into Endpoint Detection and Response (EDR) and Security Information and Event Management (SIEM) tooling, so telemetry is always flowing [5]. If a device suddenly falls behind on patching, or a user account starts pulling data from a database at three in the morning when it never has before, that is exactly the kind of anomaly the system is designed to flag in real time rather than during a quarterly audit. Many mature implementations add a further layer of user and

entity behaviour analytics on top of raw telemetry, building a statistical baseline for what "normal" looks like for a given account or service, so that deviations can be scored rather than simply matched against a fixed rule set a distinction that matters because attackers rarely repeat the exact signature of a previously known attack.

Detection alone is not much use if the response still depends on a person noticing an alert and acting on it, which is more or less what happened at Equifax, where an expired inspection certificate meant that even the detection layer itself had quietly gone blind. ZTA environments therefore lean on Security Orchestration, Automation, and Response (SOAR) tooling to close that gap. The Policy Decision Point continuously recalculates a risk score for each active session, drawing on device posture, behavioural anomalies, and threat intelligence feeds, and when a session crosses a defined threshold, the system can revoke access, terminate the session, and quarantine the endpoint without waiting for a person to intervene [6]. Some deployments use this scoring to trigger a graded response rather than an all-or-nothing one: a modest rise in risk might simply prompt a step-up authentication challenge, while a session that looks clearly compromised is cut immediately. The practical effect is that dwell time the length of time an attacker spends inside an environment before being noticed can fall from months, as it did at Equifax, to minutes, which materially limits how much damage a single compromise can do before it is contained.

1.4 Implementation Challenges

None of this comes cheaply, and the difficulties organisations run into when adopting ZTA are as much organisational as technical.

1. **Legacy technical debt.** A great deal of enterprise software was simply not built with modern authentication in mind. Systems that cannot support SAML or OIDC, or that assume flat network access, are difficult to fit into a segmented, identity-aware model without significant rework [7]. This problem is most acute in sectors running operational technology or industrial control systems, where equipment can remain in service for decades and where introducing new authentication layers risks disrupting safety-critical processes that were never designed to be interrupted for a login prompt.
2. **Friction with existing ways of working.** Tightening access inevitably slows some people down, at least initially, and overly aggressive policies have a habit of pushing users toward shadow IT rather than toward compliance. Help-desk ticket volumes commonly spike in the months following a rollout, as users who previously had standing access to shared drives or applications suddenly have to request it, and security teams have to strike a genuine balance between friction and usability rather than simply defaulting to "deny."
3. **The asset management problem.** ZTA depends on knowing, with reasonable precision, what devices, users, and data flows exist across the organisation. In a fragmented, multi-cloud estate, simply establishing that inventory let alone keeping it current turns out to be one of the harder parts of the exercise [8]. Shadow SaaS adoption, where individual teams sign up for cloud tools outside of any central procurement process, makes this worse, since a configuration management database that is even a few weeks out of date can leave entire categories of access invisible to the policy engine.
4. **Cost and phased rollout.** New infrastructure such as identity-aware proxies, SOAR platforms, and EDR tooling represents a genuine investment, on top of the time needed to retrain security staff who may be used to thinking in perimeter terms. Most organisations that succeed with ZTA treat it as a multi-year, phased programme rather than a single project, starting with the most sensitive protect surfaces first and expanding

outward, since attempting a full enterprise-wide cutover in one step tends to produce exactly the kind of user friction described above at a scale nobody can support.

5. **Federated and third-party trust.** Modern organisations rarely operate in isolation; contractors, managed service providers, and SaaS vendors routinely need some form of access into the environment, and Zero Trust's identity-centric model has to extend to those external identities as well as internal ones. Federating a policy engine across organisational boundaries, so that a vendor's device posture and identity assurance can be verified with the same rigour as an employee's, is technically harder than securing a single organisation's own estate, and is frequently the last piece of a Zero Trust programme to mature, well after internal segmentation and monitoring are already in place.

1.5 Efficacy Compared to Traditional Models

Measured against traditional perimeter defences, ZTA's advantage comes largely from a difference in starting assumptions (Figure 2). Perimeter models tend to fail badly once a single endpoint is compromised, because a foothold gained through, say, a VPN connection often opens onto a flat internal network with little standing in the way of further access. Zero Trust does not make that initial compromise less likely no model can promise that, and phishing or credential theft remains just as viable an entry point under Zero Trust as under any other architecture but it limits what an attacker can do once they are in, confining them to the identity or micro-segment they managed to breach [9]. This is reflected in the broader data on breach outcomes: organisations with mature Zero Trust deployments tend to report both lower breach costs and faster containment than those relying primarily on perimeter defences [10], largely because the same continuous verification that slows down a legitimate user's first login also slows down an attacker at every subsequent step, rather than only at the front door.

It would be an overstatement to describe Zero Trust as a solved problem, however. A policy engine is only as good as the identity system feeding it, and adversary-in-the-middle phishing kits capable of intercepting session tokens have shown that even MFA-protected logins can be undermined if the underlying credential and session-handling design is weak. Zero Trust also does very little, on its own, to stop an insider who has legitimate access and abuses it within the bounds of their normal behaviour, since the model is built to catch deviation from a baseline rather than misuse that stays inside it. What ZTA changes is the shape of the risk: instead of one breach potentially exposing an entire flat network, exposure is bounded by segment and by identity, and the time available to contain an incident is measured in minutes rather than months. The traditional model puts its effort into defending the moat; Zero Trust concentrates on defending the vault, on the assumption that the moat will eventually be crossed.

Figure 2: Efficacy — Perimeter Model vs Zero Trust Micro-segmentation

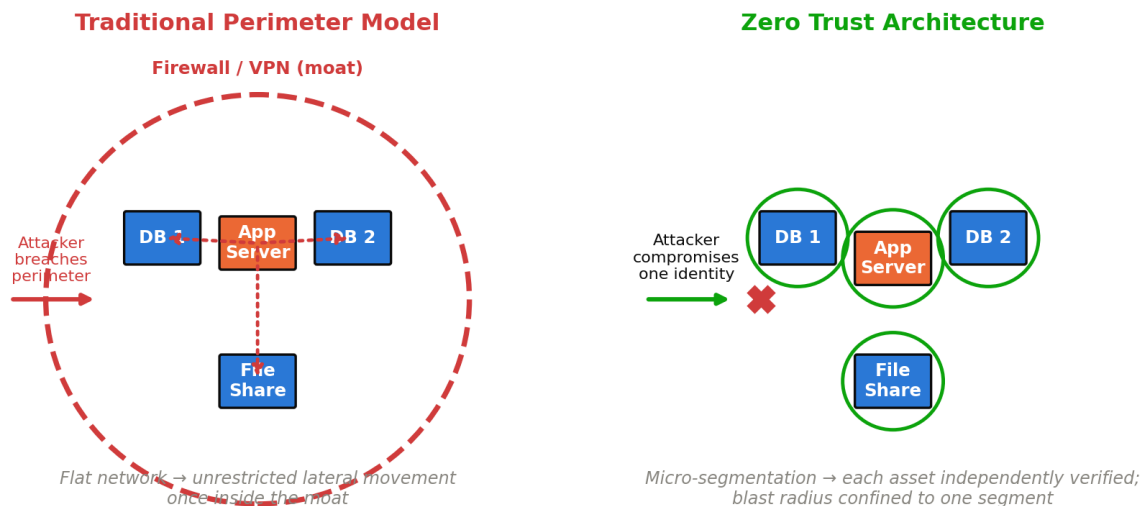


Figure 2: Traditional perimeter defence versus Zero Trust micro-segmentation.

Conclusion (Task 1)

Taken together, the four capabilities examined in this section anticipating threats through protect-surface mapping, protecting resources through explicit verification and micro-segmentation, detecting anomalies continuously rather than at login, and responding automatically through SOAR-driven playbooks describe a model that is structurally better suited to a world of cloud services, remote work, and supply-chain risk than the perimeter model it is replacing. That does not make Zero Trust a finished product that can simply be purchased and switched on. Its implementation challenges are real, particularly for organisations carrying significant legacy technical debt or operating across fragmented multi-cloud estates, and its efficacy depends heavily on the maturity of the identity and monitoring capabilities that sit underneath it. A poorly implemented Zero Trust programme, one that adds authentication friction without genuinely improving segmentation or detection, can end up costing more than it protects.

What the evidence does support is a more modest but still significant claim: where perimeter models fail catastrophically once breached, Zero Trust fails in a contained, recoverable way, and that difference in failure mode is arguably the single most important security property an organisation can buy. This is not a claim that Zero Trust removes risk from an organisation; it is a claim that it reshapes risk into something bounded and manageable, which is a more realistic goal for any security programme operating against a determined and well-resourced adversary.

Two further observations are worth drawing out before moving to the case study. First, Zero Trust is best understood as an architectural philosophy rather than a single product category, and organisations that treat it as something a vendor can sell them in a box tend to end up with fragmented tooling that enforces identity checks in one layer and network segmentation in another, with no shared risk score connecting the two. The principles set out in SP 800-207 are deliberately vendor-neutral for this reason [2]. Second, the maturity curve matters: an organisation with no existing IAM discipline, no asset inventory, and no monitoring capability is

not in a position to adopt Zero Trust in one step, and attempting to do so usually produces a programme that looks like Zero Trust on a slide deck without delivering any of the containment benefit described above. The remainder of this report applies these principles directly to the 2017 Equifax breach, treating it as a case study in exactly the kind of flat-network, low-visibility environment that Zero Trust is designed to replace, and using it to design a cybersecurity programme grounded in the principles set out above rather than in generic best practice.

References (Task 1)

- [1] E. Amoroso, **Cyber Attacks: Protecting National Infrastructure**, Woburn, MA: Butterworth-Heinemann, 2012.
- [2] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," National Institute of Standards and Technology (NIST), Special Publication 800-207, Aug. 2020.
- [3] J. Kindervag, "Build Security Into Your Network's DNA: The Zero Trust Network Architecture," Forrester Research, Nov. 2010.
- [4] R. Smith and R. Kumar, "Identity and Access Management in Cloud Environments," **IEEE Transactions on Cloud Computing**, vol. 9, no. 2, pp. 450-462, 2021.
- [5] A. O. O. Ojo, O. J. O. Ojo, and A. A. A. Ojo, "An Evaluation of Endpoint Detection and Response Systems," **IEEE Access**, vol. 8, pp. 112-125, 2020.
- [6] M. N. Al-Mhiqani et al., "A Review of Security Orchestration, Automation, and Response (SOAR) Platforms," **IEEE Communications Surveys & Tutorials**, vol. 22, no. 4, pp. 2780-2804, 2020.
- [7] D. A. Wheeler, "Software Bill of Materials (SBOM) and Supply Chain Security," **IEEE Software**, vol. 38, no. 3, pp. 10-15, 2021.
- [8] K. Zheng et al., "Micro-segmentation in Enterprise Networks: Challenges and Solutions," **IEEE Network**, vol. 34, no. 5, pp. 44-50, 2020.
- [9] U.S. Department of Defense, "DoD Zero Trust Reference Architecture," Version 2.0, Zero Trust Portfolio Management Office, Jul. 2022.
- [10] IBM Security, "Cost of a Data Breach Report 2023," IBM Corporation, Armonk, NY, 2023.

Task 2 : Case Scenario: Designing a Cybersecurity Programme based on Equifax

2.1 Introduction Context

The 2017 Equifax breach is worth dwelling on not because the technical failure was unusual, but because it wasn't. An unpatched Apache Struts vulnerability (CVE-2017-5638) sat exposed for months, and once exploited, gave attackers a foothold that went undetected for 76 days while roughly 147 million records were extracted. What makes the case useful here is how much of the failure sat above the technical layer: the CIO and CSO both reported through the Chief Legal Officer rather than to the CEO, which diluted accountability at exactly the point it mattered most; nobody held a reliable inventory of where Struts was actually deployed; an expired SSL certificate had left intrusion detection blind for months; and a flat network meant that, once inside, attackers could move around largely unimpeded [1].

The consequences extended well beyond the immediate technical incident. Congressional hearings followed, a multi-hundred-million-dollar regulatory settlement was eventually reached with U.S. federal and state authorities, and the company's leadership, including its CEO, CIO, and CSO, departed in the aftermath [2]. Those downstream costs are a useful reminder that the programme outlined below is not simply a technical exercise: getting the governance and accountability layer wrong is what turned a patchable vulnerability into an organisational crisis. It is also worth being precise about what actually went wrong from a purely technical standpoint, since the same pattern recurs across many large breaches. The vulnerable Struts component had a patch available within days of disclosure; the failure was not a lack of a fix but a lack of any reliable process for knowing which of the organisation's systems needed it applied, which is an asset-visibility failure as much as a patching failure. The programme that follows applies the Zero Trust principles discussed in Task 1 directly against these specific failures, rather than proposing a generic security uplift, and is structured around five components: organisational accountability, formal policy, strategic governance alignment, risk assessment, and compliance and incident response, each mapped explicitly back to a specific point in the breach timeline rather than presented as a generic checklist.

2.2 Organisational Structure and Accountability

One of the clearer lessons from Equifax is that a siloed reporting structure with no direct board visibility into cyber risk is itself a vulnerability. When the CISO function reports through a legal or general-counsel lens rather than directly into the executive team, security decisions tend to be framed primarily around litigation exposure rather than operational risk, which is a subtly different and less useful lens for deciding, for example, whether a patch cycle can wait another quarter. Addressing that means restructuring around executive-level accountability rather than layering more tooling onto the existing hierarchy.

Proposed Organisational Restructuring:

The Chief Information Security Officer (CISO) should sit as a peer to the CIO rather than beneath a legal or IT function, reporting directly to the CEO with an additional line into the Board's Risk Committee.

Role	Core Responsibility	Reporting Line	Justification (Addressing Equifax Failures)
Board Risk Committee	Oversight of enterprise cyber risk, budget approval, and strategic alignment.	Shareholders	Ensures cyber risk is treated as a core business risk, preventing executive ignorance of critical vulnerabilities.
CISO	Strategic leadership, policy enforcement, and enterprise security posture.	CEO (Direct) / Board (Dotted)	Prevents the conflict of interest where IT prioritises uptime or cost over security patching.
Director of Security Architecture	Designing ZTA, micro-segmentation, and network controls.	CISO	Ensures legacy flat networks are replaced with isolated, zero-trust segments to prevent lateral movement.
Director of SecOps & IR	Continuous monitoring, threat hunting, and automated incident response.	CISO	Closes the kind of 76-day dwell-time gap seen at Equifax by keeping security tooling, including SSL decryption, actively maintained and monitored.
Secure SDLC / AppSec Lead	Oversight of application vulnerabilities and code dependencies.	CISO (Collaborates w/ CIO)	Ensures open-source dependencies, such as Apache Struts, are tracked via SBOMs and patched promptly.

Table 1: Organisational Restructuring.

Figure 3: Proposed Security Governance & Reporting Structure

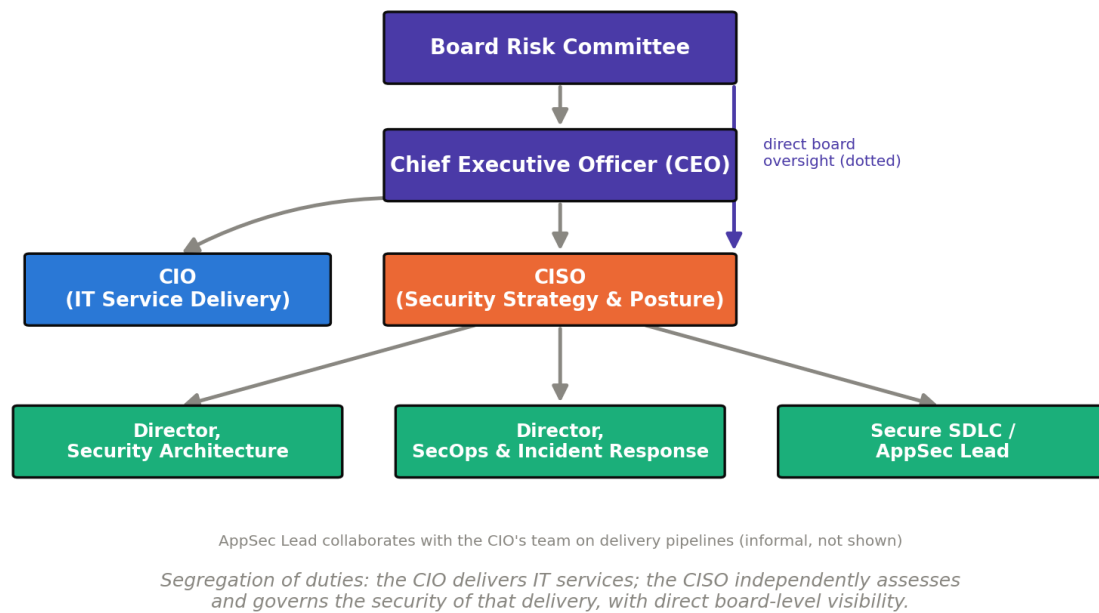


Figure 3: Proposed security governance and reporting structure.

This structure enforces a clear segregation of duties (Figure 3): the CIO remains responsible for delivering IT services, while the CISO independently assesses and governs the security of that delivery. Neither role can quietly deprioritise the other's concerns, which was arguably the deeper structural failure at Equifax, rather than the unpatched vulnerability itself. In practice this also needs a defined reporting cadence rather than an ad hoc arrangement: quarterly cyber-risk briefings to the full Board, with an escalation path that allows the CISO to convene the Risk Committee immediately outside that cycle if a critical vulnerability or active incident demands it, so that governance visibility does not depend on the next scheduled meeting happening to fall at a convenient time.

2.3 Policies and Procedures

Equifax's patch management policy existed on paper; what it lacked was any real mechanism to confirm it was being followed. The policies below are written with that gap specifically in mind, so each one pairs a mandate with a way of checking whether it is actually happening, rather than relying on an annual attestation that nobody has the time to verify in detail.

1. Enterprise Vulnerability and Patch Management Policy

- **Mandate:** Critical vulnerabilities (CVSS 9.0-10.0), particularly in internet-facing systems, must be patched or mitigated within 48 hours of discovery.
- **Enforcement:** Automated vulnerability scanners are integrated directly into IT service management (ITSM) tools. If a patch SLA is breached, the affected asset is automatically quarantined by the Zero Trust network via SOAR integration until compliance is restored.
- **Measurability:** Monthly Board reporting on "Mean Time to Patch" (MTTP) and patch compliance percentages.

2. Secure Software Development Lifecycle (SSDLC) Policy

- **Mandate:** All proprietary and third-party software must undergo Software Composition Analysis (SCA) to generate a Software Bill of Materials (SBOM) before deployment.
- **Enforcement:** CI/CD pipelines halt deployment if known, high-severity vulnerabilities exist in the code or its dependencies the exact class of issue that Apache Struts represented in 2017.

3. Cryptographic and Asset Visibility Policy

- **Mandate:** All network traffic must be decryptable by internal security tools for inspection. All SSL/TLS certificates must be centrally managed and automatically rotated 30 days prior to expiration.
- **Enforcement:** Automated Certificate Lifecycle Management (CLM) tooling a direct response to the Equifax failure where an expired certificate left exfiltrated data moving undetected.

Each of these policies is written to be auditable by a third party, not just by the team that wrote it. That distinction matters for a regulated business: an external auditor working toward an ISO/IEC 27001 certification, for instance, needs a measurable control and an evidence trail, not a statement of intent, and building that expectation into the policy from the outset avoids the retrofitting exercise that so often turns a compliance audit into a scramble.

2.4 Governance and Strategic Alignment (NIST CSF 2.0)

Tying the programme to organisational objectives means anchoring it in a recognised framework rather than an ad hoc list of controls, and NIST CSF 2.0 is a reasonable choice here because it maps cleanly onto both the Equifax failures and the Zero Trust principles already discussed (Figure 4). The 2024 update to the framework is itself instructive: CSF 2.0 added GOVERN as a standalone function sitting above the original five, precisely because earlier versions treated governance as implicit rather than as something to be assessed on its own terms [3]. That change mirrors the central lesson of the Equifax case almost exactly a strong technical control set is of limited value if the reporting structure sitting above it cannot act on what those controls reveal.

NIST CSF 2.0 Core Function	Strategic Application to Proposed Programme (Equifax Context)
GOVERN (GV)	Establishes the CISO-to-Board reporting structure and integrates cyber risk into enterprise risk management (ERM), so leadership treats protecting PII as inseparable from business survival and market trust.
IDENTIFY (ID)	Implements dynamic asset discovery and SBOM generation. Equifax's core failure here was not knowing where Struts was deployed; this function exists to close exactly that gap.

PROTECT (PR)	Deploys Zero Trust micro-segmentation, Data Loss Prevention (DLP), and multi-factor authentication (MFA), so critical databases stay protected even if a web portal is compromised.
DETECT (DE)	Uses AI-assisted EDR and network traffic analysis, including active SSL inspection, to catch anomalies before they turn into large-scale exfiltration.
RESPOND (RS)	Uses automated SOAR playbooks to isolate compromised nodes almost immediately, supporting both legal compliance and crisis communication needs.
RECOVER (RC)	Maintains immutable backups and rehearsed restoration procedures, so a contained incident does not turn into extended operational downtime.

Table 2: Strategic Application to Proposed Programme.

Figure 4: NIST CSF 2.0 Core Functions Applied to the Programme

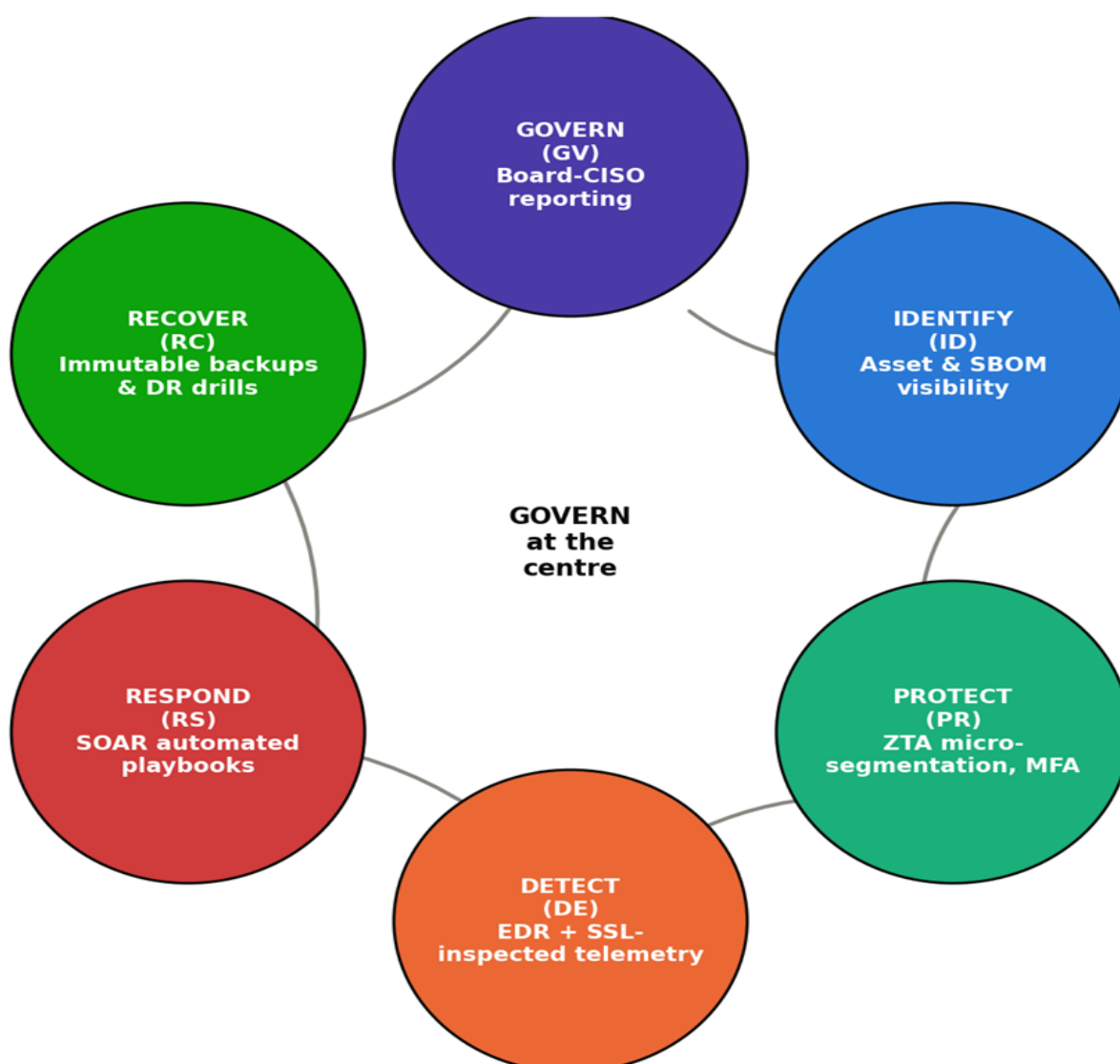


Figure 4: NIST CSF 2.0 core functions applied to the programme.

The value of mapping the programme this way is that cybersecurity stops being read purely as an IT cost line. Framed through GOVERN, IDENTIFY, PROTECT, DETECT, RESPOND, and RECOVER, the same controls read instead as business continuity and trust measures, which is a more accurate description of what they are actually doing, and it gives the Board a common vocabulary for asking questions about the programme without needing to understand the underlying technology in detail.

2.5 Risk Management and Assessment

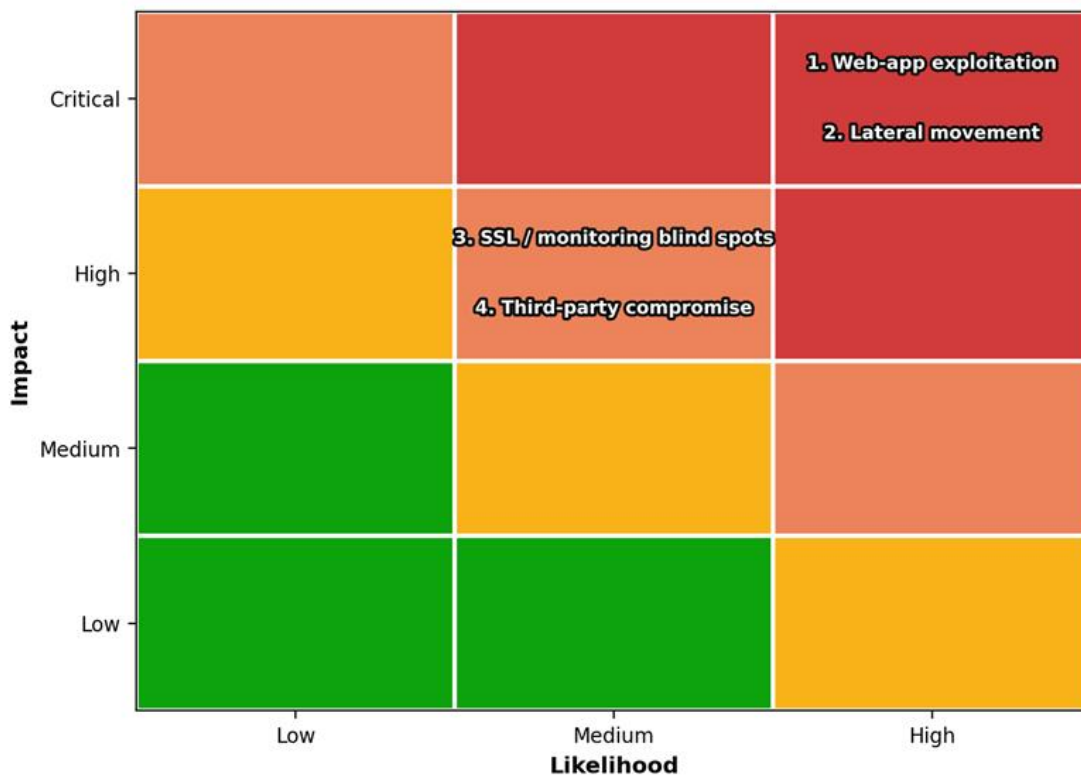
A risk assessment is only useful if it forces some prioritisation, so the matrix below focuses specifically on the vulnerability classes that were actually exploited or left unaddressed at Equifax, rather than attempting to be exhaustive (Figure 5). The scoring approach follows the standard likelihood-times-impact logic found in frameworks such as NIST SP 800-30, with impact assessed against confidentiality of PII, regulatory exposure, and operational continuity, and likelihood assessed against how directly a given weakness maps onto a known, exploited attack pattern rather than a purely theoretical one.

High-Level Risk Assessment Matrix.

Risk Scenario	Vulnerability / Weakness	Impact	Likelihood	Risk Level	Mitigation Strategy (Zero Trust Focus)
Exploitation of Web Facing Application	Unpatched dependencies (e.g., Apache Struts), lack of WAF.	Critical (Data Loss, Regulatory Fines)	High	CRITICAL	Automated SCA in CI/CD; strictly enforce 48-hour patch SLA; deploy Next-Gen WAF.
Lateral Movement Post-Breach	Flat internal network architecture; excessive user privileges.	Critical (Massive DB exfiltration)	High	CRITICAL	Implement ZTA micro-segmentation; enforce Least Privilege and JIT access.
Blind Spots in Network Traffic	Expired SSL certificates on inspection devices; lack of continuous monitoring.	High (Dwell time increases)	Medium	HIGH	Centralized Certificate Management system; automated alerting for expiring certs.
Supply Chain / Third-Party Compromise	Unvetted vendor access to internal systems.	High	Medium	HIGH	Extend ZTA policies to third parties; require strict Identity-Aware Proxy access.

Table 3: Risk Assessment Matrix.

Figure 5: Risk Assessment Matrix — Equifax-derived Risk Scenarios



1 Exploitation of internet-facing web app • 2 Lateral movement post-breach • 3 Blind spots in network traffic • 4 Supply-chain / third-party compromise
 Risk level
 Medium High High+ Critical

Figure 5: Risk assessment matrix for Equifax-derived risk scenarios.

Mitigation Rationale: The two highest-priority items web application exploitation and lateral movement sit at Critical because they map directly onto how the actual breach unfolded: an unpatched public-facing vulnerability followed by largely unrestricted movement across a flat network. Certificate blind spots and third-party exposure are rated High rather than Critical, but not by a wide margin, since Zero Trust's own "assume breach" logic means even secondary weaknesses need a credible mitigation path in case the primary defences fail. Risk ratings here are also treated as living values rather than a one-off exercise: the matrix is intended to be reassessed at least quarterly, and immediately after any material change to the technology estate, such as a new third-party integration or a newly disclosed critical CVE affecting deployed software.

2.6 Compliance, Incident Response, and Protection Controls

Regulatory Alignment and Breach Notification

Any programme operating across jurisdictions has to reckon with an uneven regulatory landscape GDPR, CCPA, and, depending on where the organisation operates, newer regimes such as DORA and NIS2 that extend similar expectations to operational resilience and critical-infrastructure sectors [4], [5].

- **Protocols:** In practice this means the Incident Response (IR) plan needs a pre-agreed legal escalation path. Confirmed PII breaches should trigger notification to the legal department within one hour, with regulators such as the ICO informed inside the mandatory 72-hour GDPR window. That is a deliberate contrast with Equifax, where public disclosure took months rather than days.

Incident Response and Disaster Recovery (DR)

- **IR Planning:** Automated playbooks matter more here than documentation. When EDR flags anomalous behaviour, SOAR scripts should isolate the endpoint immediately rather than waiting for a ticket to be picked up [6].
- **DR Planning:** Disaster recovery planning is kept deliberately unglamorous: regular table-top exercises involving C-suite executives, rehearsing both technical restoration and the crisis communication that has to accompany it. These exercises are most useful when run at least twice a year and when at least one session per year assumes the primary incident commander is unavailable, since a plan that only works when the most experienced person in the room is present is not really a plan.

Technical Safeguards and Dwell Time Reduction

Underneath the policy layer, a specific set of controls does most of the practical work:

- **Continuous Monitoring & Automated Scanning:** Vulnerability scanners run constantly, tied to threat intelligence feeds so newly disclosed CVEs are prioritised as soon as they appear.
- **Endpoint Detection & Response (EDR):** Deployed broadly enough to catch file-less malware and unusual execution patterns through behavioural analytics, not just signature matching.
- **Encryption:** Data is encrypted both in transit (TLS 1.3) and at rest (AES-256). Database-level tokenisation goes a step further, so stolen administrator credentials alone are not sufficient to make off with usable PII.
- **Strong Authentication:** Phishing-resistant MFA, such as FIDO2 hardware keys, is mandatory for all administrative access, rather than SMS or app-based codes that remain vulnerable to interception.

Justification: Each control here maps onto a specific stage of the Equifax breach rather than functioning as a generic best practice (Figure 6). Encryption at rest and micro-segmentation together make lateral movement to sensitive databases significantly harder even after an initial compromise. Automated certificate management and continuous monitoring address the exact blind spot that let the 2017 breach run undetected for as long as it did. And EDR combined with SOAR is what actually compresses dwell time from the 76 days Equifax experienced down to a matter of minutes which is ultimately what separates a contained incident from an enterprise-wide catastrophe.

Conclusion (Task 2)

Equifax was not undone by a single missed patch so much as by visibility, accountability, and architecture failing to work together, and the programme set out in this section has been built to address each of those failure modes individually rather than treating the incident as a single technical lapse. An executive-level CISO with direct board access closes the accountability gap; measurable, auditable policies close the gap between what was documented and what was actually enforced; NIST CSF 2.0 gives the whole programme a shared vocabulary that connects technical controls back to business continuity; the risk matrix keeps investment focused on the vulnerability classes that actually caused the breach rather than a generic list; and the incident-response and technical-control layer is designed specifically to compress the 76-day dwell time that allowed the original exfiltration to continue for as long as it did.

None of this guarantees a breach never happens again nothing does, and a determined attacker with a novel technique will always have some chance of finding a foothold. What the programme changes is the default assumption an organisation operates under, from trusting whatever sits inside the perimeter to verifying everything, continuously, in line with the Zero Trust principles established in Task 1. If a comparable vulnerability were exploited under this programme, the expected outcome is a single contained segment and a same-day detection, rather than a 147-million-record catastrophe unfolding quietly over 76 days.

References (Task 2)

- [1] M. M. Ahmad, "Analyzing the Equifax Data Breach: Vulnerabilities, Exploitations, and Mitigation Strategies," **International Journal of Information Security**, vol. 19, no. 3, pp. 320-335, 2020.
- [2] Committee on Oversight and Government Reform, "The Equifax Data Breach," U.S. House of Representatives, Majority Staff Report, 115th Congress, Dec. 2018.
- [3] National Institute of Standards and Technology, "The NIST Cybersecurity Framework (CSF) 2.0," NIST CSWP 29, Feb. 2024.
- [4] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," **Official Journal of the European Union**, Apr. 2016.
- [5] European Parliament and Council, "Regulation (EU) 2022/2554 on Digital Operational Resilience for the Financial Sector (DORA)," **Official Journal of the European Union**, Dec. 2022.
- [6] M. N. Al-Mhiqani et al., "A Review of Security Orchestration, Automation, and Response (SOAR) Platforms," **IEEE Communications Surveys & Tutorials**, vol. 22, no. 4, pp. 2780-2804, 2020.
- [7] K. Zheng et al., "Micro-segmentation in Enterprise Networks: Challenges and Solutions," **IEEE Network**, vol. 34, no. 5, pp. 44-50, 2020.
- [8] IBM Security, "Cost of a Data Breach Report 2023," IBM Corporation, Armonk, NY, 2023.
- [9] D. A. Wheeler, "Software Bill of Materials (SBOM) and Supply Chain Security," **IEEE Software**, vol. 38, no. 3, pp. 10-15, 2021.
- [10] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," National Institute of Standards and Technology (NIST), Special Publication 800-207, Aug. 2020.

Appendix

Appendix A: Equifax Attack Vector Diagram (Conceptual)

1. **Initial Vector:** Attacker probes Equifax Dispute Portal.
2. **Exploitation:** CVE-2017-5638 (Apache Struts) triggered via malicious HTTP header.
3. **Foothold:** Web shell deployed on application server.
4. **Lateral Movement:** Attacker accesses flat internal network, mounts internal file shares containing unencrypted application credentials.
5. **Data Exfiltration:** Attacker accesses 51 different databases over 76 days. Exfiltration goes unnoticed due to expired SSL certificate on the intrusion detection system (IDS) appliance.

Appendix B: Zero Trust Mitigation Mapping

- **Block Initial Vector:** WAF filters malicious headers. Automated patching prevents the vulnerability from existing.
- **Block Foothold:** EDR detects anomalous web shell execution and blocks it.
- **Block Lateral Movement:** Micro-segmentation strictly isolates the web server from the broader internal network.
- **Block Exfiltration:** SOAR detects abnormal data transfer volumes and severs the connection; automated SSL management ensures IDS maintains visibility.

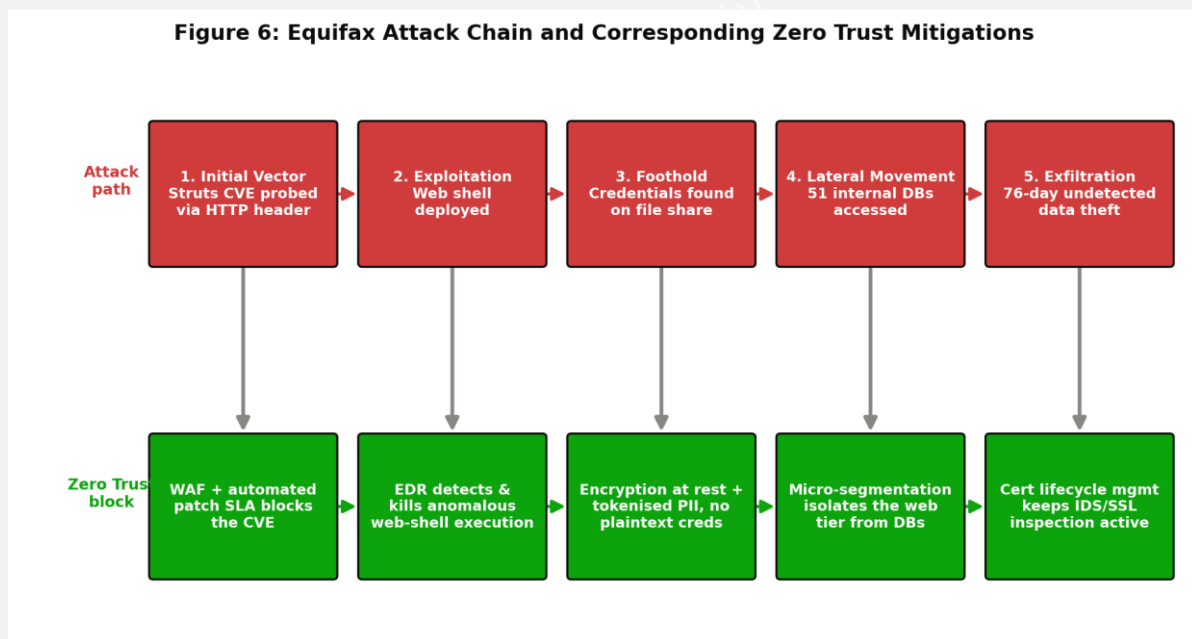


Figure 6: Equifax attack chain mapped against Zero Trust mitigations at each stage.