A Systematic Analysis of the Capital One Data Breach: Critical Lessons Learned

SHAHARYAR KHAN, ILYA KABANOV, YUNKE HUA, and STUART MADNICK,
Sloan School of Management, Massachusetts Institute of Technology

The 2019 Capital One data breach was one of the largest data breaches impacting the privacy and security of personal information of over a 100 million individuals. In most reports about a cyberattack, you will often hear that it succeeded because a single employee clicked on a link in a phishing email or forgot to patch some software, making it seem like an isolated, one-off, trivial problem involving maybe one person, committing a mistake or being negligent. But that is usually not the complete story. By ignoring the related managerial and organizational failures, you are leaving in place the conditions for the next breach. Using our Cybersafety analysis methodology, we identified control failures spanning control levels, going from rather technical issues up to top management, the Board of Directors, and Government regulators. In this analysis, we reconstruct the Capital One hierarchical cyber safety control structure, identify what parts failed and why, and provide recommendations for improvements. This work demonstrates how to discover the true causes of security failures in complex information systems and derive systematic cybersecurity improvements that likely apply to many other organizations. It also provides an approach that individuals can use to evaluate and better secure their organizations.

CCS Concepts: • **Security and privacy → Firewalls**;

Additional Key Words and Phrases: Capital One breach, cybersafety, cybersecurity, privacy, STAMP

ACM Reference format:
Shaharyar Khan, Ilya Kabanov, Yunke Hua, and Stuart Madnick. 2022. A Systematic Analysis of the Capital One Data Breach: Critical Lessons Learned. *ACM Trans. Priv. Sec.* 26, 1, Article 3 (November 2022), 29 pages.
https://doi.org/10.1145/3546068

1 INTRODUCTION

Precipitated by the exposure of business processes and personal information to the public internet (either due to increased connectivity of on-premise data centers to the internet or displacement of business services to the cloud), there has been a marked increase in the number of data hacks in the past decade [1]. The frequency of these hacks has reached a point where they rarely illicit any significant surprise or shock from the general public.

This research was supported, in part, by funds from the corporate members of Cybersecurity at MIT Sloan: the Interdisciplinary Consortium for Improving Critical Infrastructure Cybersecurity.
Authors' address: S. Khan, I. Kabanov, Y. Hua, and S. Madnick, MIT Sloan School of Management 100 Main Street Room E62-364 Cambridge, Mass 02142; emails: {shkhan, ikabanov, huay, smadnick}@mit.edu.

2471-2566/2022/11-ART3 $15.00
https://doi.org/10.1145/3546068

On the surface, the Capital One data breach announced on July 29, 2019 [2], is just "another" hack; a misconfigured web application firewall enabled exfiltration of sensitive private credit card application data. The size of the data breach (at 106 million affected individuals in the United States and Canada) makes it one of the worst data breaches (in terms of number of records exposed) of the previous decade along with Equifax, Target, Marriott, and others [1]. However, more than the scale of the breach, this particular cyberattack is a fascinating case study because of several reasons.

For one, unlike other major breaches, a lot of the technical details about the method of exploitation in this data breach were revealed by the attacker herself in public forums and blogs [3]; this has enabled piecing together a comprehensive narrative of the attack. Second, unlike Equifax, which operated a legacy data infrastructure with poor patch management [4], Capital One operated a state-of-the-art cloud infrastructure using encryption and tokenization. In fact, Capital One was regarded as one of the most cloud-savvy enterprises at the time of the breach; it was an early and vocal advocate for **Amazon Web Services (AWS)**, and their transition to the cloud was touted as an example for others to emulate. Third, the attack did not include any significantly novel technique (i.e., a zero-day exploit), but rather exploited a number of well-understood vulnerabilities, including a **Server-Side Request Forgery (SSRF)** [5] and a weakness in the AWS EC2 service infrastructure [6]. Fourth, the compromised data was stored on the public cloud and was supposedly encrypted; however, the particular circumstances surrounding this breach enabled the attacker to decrypt the data as well [2]. Considered together, this means that this attack was successful, not because of its novelty, but because of a number of control failures at various levels of the organization, any one of which, if adequately enforced, would have been able to prevent the attack or at the very least would have been able to limit the scale of the breach. This situation has been found in other major attacks as well, such as Equifax, and therefore, it is important to learn from them to prevent further attacks to our organizations.

In their analysis of the Capital One data breach, Neto et al. [7] attempt to identify the various controls that were missing, inadequate, or ineffective that enabled this cyberattack to be successful. Although pertinent, their focus is on identifying the technical controls relevant to proximate events immediately preceding the loss [7]. As Leveson [8] argues "*the foundation for an accident* [or in this case, the cyber breach] *is often laid years before*"; the event (or misconfiguration) simply acts as a trigger for the loss—had it not been for this event, something else would have likely resulted in a similar outcome.

In other words, incorrect/inadequate decisions at higher levels of the system, based on incorrect "beliefs" or "assumptions" about the system or the environment, create systemic vulnerable conditions that may take years to manifest but ultimately result in losses as a result of seemingly insignificant trigger-events. It is this deeper level of understanding that this analysis tries to uncover. In this analysis, we employ a systems-thinking based approach to understand why certain controls or decisions were ineffective, missing, or inadequate and why the issuance (or lack thereof) of these control actions and decisions may have made sense at the time, given the context in which they were given, but in retrospect, appear to be deeply flawed.

Based on our holistic analysis, we provide concrete, actionable recommendations for organizations to improve their security posture. These include, among other things, advice for management to match the pace of new technology adoption with maturity of their risk management practices, implementation of more rigorous risk management processes, and recognition of their organization's changed roles and responsibilities when operating in the cloud—which includes revisiting the *shared-responsibility model*. We also recommend active involvement of the Board in security matters along with inclusion of CISOs in board meetings to improve the board's understanding of cybersecurity risks so they can actively prioritize resources to address cyber risks. Based on our analysis, the **cloud service provider (CSP)** also has an important role to play; CSPs must try to

reduce complexity in their platform and ensure security is simplified and prioritized. CSPs must also prioritize fixing of vulnerabilities in their platforms over release of new services and features. We also identified a lack of regulatory oversight of CSPs along with layering of multiple uncoordinated security requirements by various regulatory bodies as causal factors for ineffectiveness of some security controls.

At the operational level, we advise that the IT and InfoSec team's must also understand their roles and responsibilities under the *shared-responsibility model* and must reinforce proven and mature security practices when operating in the cloud, such as *trust-but-verify, least privilege, defense-in-depth*, and so on. In addition, InfoSec teams must institute robust application review, vulnerability management, and intrusion detection processes that transcend a *"check-box"* mentality. Finally, our most important recommendation is that organizations must think of security as a *system* problem, focusing not only on security of individual components, but also those vulnerabilities that emerge from interactions between components such as latent flaws resulting from poor organizational structures and missing informational feedbacks.

2 CYBERSAFETY METHOD OF ANALYSIS

To better understand the causal factors for the Capital One data breach, we utilize the Cybersafety method [9]. Cybersafety is a technique, similar to **CAST (Causal Analysis using Systems Theory)** and **STPA (System Theoretic Process Analysis)** methods of analyses, that is inspired by the accident-causality framework proposed by Leveson [10] known as **System-Theoretic Accident Model and Processes (STAMP)**. There are intrinsic differences between cyber incidents and accidents, where a cyberattack is caused by deliberate malicious actions leveraging flaws within a system while an accident has some element of randomness associated with it [10].

The Cybersafety method was first used by Salim & Madnick [11] to analyze the TJX cyberattack. It was later also used by Kabanov & Madnick [4] to analyze the Equifax cyberattack. The method was further extended to analyze complex industrial control systems *ex-ante* to identify cyber-vulnerabilities and mitigation strategies to protect against not-yet-seen cyberattacks by Khan & Madnick [9].

Whereas traditional causality models attempt to identify root causes for accidents such as individual component failures or human errors cascading into accidents, the Cybersafety approach tries to identify flaws within the design of the system (both *process/mental model flaws*[1] as well as structural flaws) that create the conditions necessary for the event to occur.

In the Cybersafety method, the system is perceived as a number of hierarchical controllers or decision-makers, each enforcing safety and security constraints on the system; it is the violation of these constraints that moves the system into unsafe states, leading to losses. In other words, in addition to identifying "what went wrong," the Cybersafety method attempts to uncover why it made sense for the various controllers and decision-makers to undertake the incorrect actions and decisions at the time within a certain context and how the indirect interactions between the various controllers ultimately resulted in a loss of control of the system resulting in the catastrophic event [9]. The basic steps in the Cybersafety method are summarized as follows:

- **Step 1:** Collect the necessary information about the incident (including proximate events) as well as the system, including the physical system and processes involved in the loss.
- **Step 2:** Develop a hierarchical functional control structure to model the controllers and their interactions that together are intended to enforce safety and security constraints on the system (i.e., control the hazards).

[1] A *Process model* refers to a state or model of the process that a controller uses to take a control action to keep the process within certain defined limits to ensure safety/security; in humans, this is referred to as a *mental model.*

- **Step 3:** Examine the various controllers in the control structure to determine why they were not effective in preventing the loss. This includes identifying the roles and responsibilities of the various controllers and determining what controls were inadequate, missing, or ineffective.
- **Step 4:** Identify flaws in the control structure as a whole. This includes understanding the context and environmental conditions that the system was subject to that caused the controllers to behave as they did, along with determining the assumptions and deep-rooted process/mental model flaws that made the controls ineffective.
- **Step 5:** Create recommendations for improvements to the safety control structure to prevent a similar loss in the future.

In this article, we will apply the Cybersafety method to the Capital One cyber incident. We will start by piecing together the events that led to the breach, including a description of the actual event.

3 THE INCIDENT

We employ three *types* of information sources to perform the Cybersafety analysis on the Capital One data breach. First, we examine *official sources*, including the official disclosure of the data security incident by Capital One [12], the official court filing of the indictment in the US District Court at Seattle [3], and subsequent court proceedings along with the report by the **Office of the Comptroller of the Currency (OCC)** [13] to assess penalties against Capital One. Second, we examine public forums, security blogs, news articles, and so on, [6, 14–19] written by security experts. Although usually speculative in nature and sometimes even contradictory, these articles provide useful context about the technical details that, in conjunction with the official reports, enable piecing together a coherent story about the breach. Finally, we utilize *informed judgement* to identify potential flaws in assumptions and mental models that might have led to error-prone decision-making and enforcement of ineffectual/inadequate controls.

The official announcement by Capital One [12, 20] on July 29 stated that a data security incident resulted in the disclosure of personal data of approximately 100 million individuals in the United States and approximately 6 million in Canada. Further details revealed that the incident was caused by the unauthorized access to a Capital One data server by a former software engineer of **Amazon Web Services (AWS)**, Paige A. Thompson—the primary suspect in the case [20]. The attacker obtained personal information that the bank routinely collected for its credit card applications, including names, addresses, zip codes/postal codes, phone numbers, email addresses, dates of birth, and self-reported incomes [11]. Beyond the credit card application data, portions of credit card customer data such as customer status data (e.g., credit scores, credit limits, balances, payment history, contact information), fragments of transaction data, Social Security numbers, and some linked bank account numbers of Capital One's credit card customers were obtained by the intruder as well [11].

3.1 Historical Context

The events leading up to the breach can be traced back to the launch of Capital One's Cloud Strategy in 2014 when George Brady joined the bank as its Chief Technology Officer [21]. Around the same time, Capital One took some very bold and unorthodox decisions for a company operating in such a highly regulated industry (i.e., financial industry) in support of its cloud strategy. These included commitment to open-source technology, adoption of the agile development philosophy, and transition to the public cloud (i.e., AWS as opposed to private cloud). This strategic transformation was accompanied with an aggressive acquisition of tech talent. In addition, the bank actively sought public perception of a forward-looking technology company as opposed to a

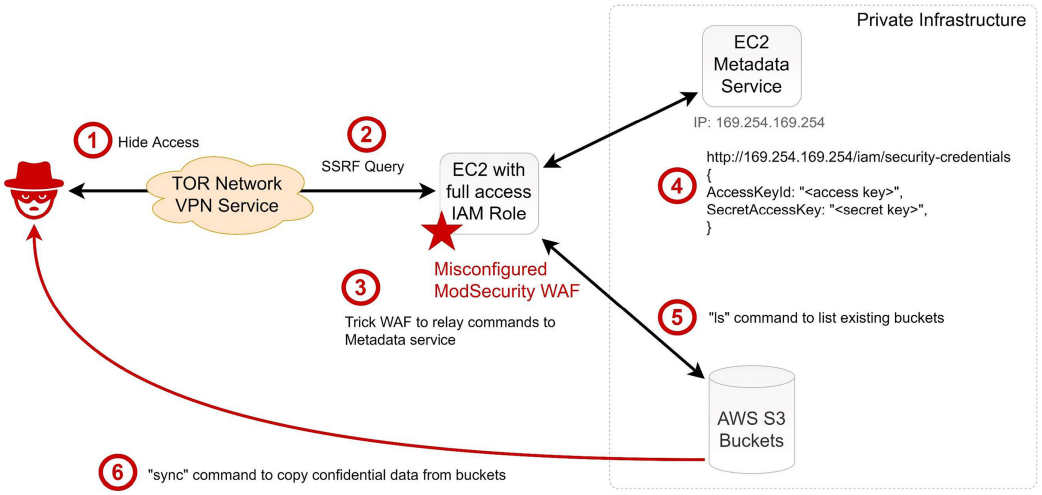

Fig. 1. Capital one cyberattack [7].

traditional bank, where the focus was on rapidly developing new capabilities to enhance customer experience.

The 2019 cyber breach was not the only data breach that the company suffered during this strategic transformation; albeit, it was by far the most impactful. In fact, twice before, once in [22] and again in [23], the company suffered minor data breaches involving insider personnel.[2] In contrast, the 2019 attack was a highly sophisticated attack that directly targeted the cloud infrastructure. We will now present the technical details of the 2019 cyber breach.

3.2 Technical Details

Some of the technical details of the data breach have been presented in the official indictment against the attacker [3]. Other details have been discussed by several security experts [5, 7, 19] in security blogs, public forums, and news articles. The main steps in the attack are briefly summarized below and illustrated in Figure 1:

(1) The attacker used anonymizing services (such as TOR and VPN service provider *IPredator* end nodes) to access Capital One's cloud network between March 12 and July 19, 2019.[3] This was confirmed, *ex post facto,* by Capital One after reviewing their network data logs [3]. The intrusions, querying of subsequent backend resources, and exfiltration of data, all remained undetected by the intrusion detection and monitoring systems in place.

(2) The next step in the attack has been the source of some controversy and speculation. One hypothesis is that the Web Application Firewall (later revealed to be the open-source Modsecurity WAF) was in fact the target of the attack. Krebs [19] states that the "*misconfiguration of the WAF allowed the intruder to trick the firewall into relaying requests to a key back-end resource on the AWS platform.*" This position is challenged by Folini [24] in a tweet—co-author of the handbook on Modsecurity, questioning how the WAF could be (mis)configured to relay commands to the backend metadata service. Further technical analysis by Walikar [5] presents a slightly different hypothesis; the attacker bypassed the

[2]In 2014, an insider stole more than $100,000 from 2 customers [22], while in [23], an insider accessed sensitive data for around 600 customers.

[3]Anonymizing services prevent revealing the source IP address of the malicious actor.

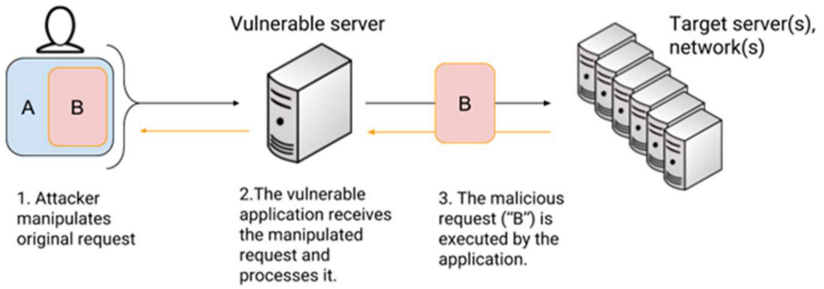

Fig. 2. Typical exploitation of a server-side request forgery vulnerability [25].

WAF due to a misconfiguration and exploited a **Server-Side Request Forgery (SSRF)** vulnerability in a web application behind the WAF. In the second hypothesis, the misconfiguration of the WAF enabled reaching the SSRF-vulnerable application that was then used to relay messages to the back-end resource.

Note that SSRF is a well-understood attack method [5]; the simplified diagram in Figure 2 shows the key steps involved in exploiting an SSRF vulnerability. The basic idea is to gain access to data (that one does not have access to) by persuading the server that has access to the sensitive data to make the request on the attacker's behalf—either by spoofing URLs or making network/HTTP requests via data input fields in the vulnerable application [5].

In its response to Senators Ron Wyden and Elizabeth Warren [26], AWS officially dismissed the second hypothesis and instead stated that the attacker engaged in an *open reverse proxy attack*, rather than SSRF. The subtle distinction, as explained by AWS being, that in a *reverse proxy attack* a misconfigured application is leveraged to query internal networking resources for sensitive data, while in the latter a web application with an SSRF vulnerability is targeted [26].

Note that similar to *SSRF, reverse proxy attacks* have also been known to the security community since at least 2016 [27, 28]. For context, Modsecurity supports two deployment options—*embedded* and *reverse proxy* deployment modes [29]. In a demonstration of the *reverse proxy attack*, security researchers [30] argue that the identity of the reverse proxy server is not necessarily relevant to the attack—a range of reverse proxy servers (such as WAFs, Nginx, Squid, etc.) could be misconfigured to result in such a breach. Kaushik [29] presents a walk-through of a *reverse proxy attack* using a hypothetical web application and a misconfigured *Nginx* reverse proxy server.

Given the official statement by AWS, the known deployment options for Modsecurity, and the known attack vectors against reverse proxies, we take the position that the Modsecurity firewall was in fact misconfigured in the reverse proxy deployment mode and was the initial target of the attack.

(3) In the case of Capital One, the Modsecurity was running on an AWS **Elastic Compute Cloud (EC2)**[4] instance. After discovering the misconfigured WAF, the attacker queried the API[5]-handling service known as AWS metadata service. The AWS metadata service returned information about the **Identity and Access Management (IAM)** role that was attached to the EC2 instance along with temporary credentials (access token) for the role. The role, titled *ISRM-WAF-Role,* attached to this instance appears to have excessive privileges allowing the listing and accessing of S3 buckets with the sensitive data.

[4]EC2 is part of Amazon Web Services that allows users to rent virtual computers on which to run their own computer applications.
[5]API stands for application programming interface.

(4) This privilege was used by the attacker to list the buckets and download them locally. Although the data was encrypted, the intruder was able to decrypt the data as well. Capital One announced "*We encrypt our data as a standard. Due to the particular circumstances of this incident, the unauthorized access also enabled the decrypting of data"* [31]. Based on this announcement, it is speculated that the role attached to the instance (*ISRM-WAF-Role)* also allowed decryption of data (since it most likely had *kms:decrypt* privilege as well). As a consequence, nearly 30 GB of Capital One credit application data (or 700 S3 buckets) was downloaded by the attacker.

To summarize, the attack was successful due to five distinct control failures; (1) existence of a misconfigured reverse proxy (Modsecurity WAF), (2) weakness of the cloud infrastructure that enabled querying the metadata service and provided temporary credentials to the attacker, (3) the existence of an over-provisioned IAM role that granted access to the S3 storage buckets, (4) the ineffectiveness of the encryption method used, and finally, (5) the ineffectiveness of the intrusion detection and monitoring systems in place. Now that we understand the proximate events and the technical details of the attack, we will apply the Cybersafety method to understand what controls should have been in place to prevent such an adverse outcome and why those controls were ineffectual or inadequate.

4 CYBERSAFETY ANALYSIS OF THE INCIDENT

4.1 The System, Hazards, and Safety & Security Constraints

As mentioned earlier, "*the foundation for an accident* [or in this case, the cyber breach] *is often laid years before"* [6]. The technical details of the Capital One data breach reveal that multiple safety/security constraints were violated that enabled the attack to be successful.

For the purpose of this analysis, we define the system boundary as anything that is within the control of Capital One as well as anything that directly enforces control/influence over Capital One. Next, we identify system-level hazards; these are system conditions that, if not controlled, result in violation of system safety and/or security constraints that ultimately translate into losses [6]. To methodically identify system-level hazards and constraints, we utilize the same approach that was employed by Kabanov & Madnick [4] in defining the hazards for the Equifax data breach, i.e., use the Cyber Kill Chain framework introduced by Lockheed Martin to identify hazards [32]. The Cyber Kill Chain consists of seven phases associated with the typical steps taken by an adversary to launch a cyberattack, including (*1) Reconnaissance, (2) Weaponization, (3) Delivery, (4) Exploitation, (5) Installation, (6) Command and Control, (7) Actions on Objectives* [32].

During the analysis of the attack on Capital One, we identified seven significant hazards associated with various phases of the Cyber Kill Chain. Note that based on the system boundary defined above, the *Reconnaissance* and *Weaponization* phases of the Cyber Kill Chain are beyond the purview of the system's control (instead, these two phases are under control of the attacker) and hence not considered as *hazards* in the analysis. The *Installation* phase is also not applicable, since this breach does not involve installation of a backdoor or malware per se; rather, the attacker exploited a misconfiguration to gain access and exfiltrated sensitive data. Table 1 presents each of the six identified system-level hazards along with the associated phase of the Cyber Kill Chain and the violated system-level constraint that enabled the attack.

Examining the[6] hazards in the context of the technical details provided earlier, we can begin to understand the multiple violations of constraints that made the attack successful. Hazard **H-1**

[6]Principle of least privilege means that in a particular abstraction layer of a computing environment, every module (such as a process, a user, or a program, depending on the subject) must be able to access only the information and resources that are necessary for its legitimate purpose.

Table 1. System-level Hazards and Constraints

Cyber Kill Chain Phase	#	System-level Hazard	Constraint Violated
Delivery	H-1	System does not have adequate protection against delivery of an exploit (i.e., inadequate protections in place to prevent delivery of SSRF, reverse proxy attack, etc.)	System must have adequate protections against delivery of SSRF, reverse proxy attacks
	H-2	System has inadequate intrusion detection and monitoring in place, i.e., system does not detect an intrusion by an attacker and does not monitor IAM API calls or reading/writing of sensitive S3 buckets	System must have adequate intrusion detection and monitoring systems in place to detect anomalous behavior
Exploitation	H-3	System is operated with an exploitable vulnerability or a misconfigured resource that allows access to backend resources	System must not be operated with an exploitable vulnerability or a misconfigured resource that allows access to backend resources
Command & Control	H-4	System access control is overly permissive beyond least privilege	System access control must follow the principles of least privilege
	H-5	System does not prevent unauthorized user from harvesting credentials and establishing control over resources	System must have an adequate mechanism to protect access to credentials
Action on Objectives	H-6	System does not adequately encrypt sensitive data	System must adequately encrypt sensitive data

refers to the lack of protection against *reverse proxy* attacks, both in the underlying cloud infrastructure as well as in the deployment of services by Capital One. This hazard refers to the respective responsibilities of each entity under the *shared-responsibility model* (namely, the cloud service provider and Capital One) and their failure in implementing the necessary protections against such attacks. Hazard **H-2** refers to the lack of adequate intrusion detection and monitoring. The attacker, after discovering the misconfigured firewall, made IAM API calls in addition to reading and writing sensitive data from S3 buckets, but none of these actions raised any alarms/alerts for the InfoSec team.

Hazard **H-3** points to the existence of a misconfiguration in the WAF and the decision to operate with that misconfiguration; had the WAF been configured properly, the attack would have been thwarted at its initial stage.

Hazard **H-4** refers to another control failure—violation of least privilege principle. The attacker was able to gain access to and establish control over sensitive data, because the role associated with the compromised EC2 instance had been granted excessive privileges, i.e., it is *believed* that the WAF did not need read/write access to the S3 buckets that stored sensitive information, however, the role associated with the WAF instance gave it those privileges.

Meanwhile, Hazard **H-5** refers to the attacker being able to query the AWS metadata service to harvest the IAM role and associated key that is then leveraged to access S3 buckets and decrypt

data; this hazard refers to a weakness in the underlying design of the authentication process in cloud infrastructure that inherently *trusts* API calls from a compromised resource. Finally, Hazard **H-6** refers to use of inadequate encryption techniques to store sensitive data.

Each one of these hazards refers to significantly different control failures, any one of which, if adequately enforced, would have been able to thwart the cyberattack on its own or, at the very least, would have been able to limit the extent of the damage. The "*misconfigured firewall*" is just the tip of the proverbial *iceberg*—Capital One's data breach was more than a component failure; rather, it was a systems failure where several components failed to function as intended, resulting in the loss.

4.2 Hierarchical Functional Control Structure

In the previous subsection, we identified several system-level hazards whose lack of control resulted in the breach. In this subsection, we identify controllers or decision-makers who were responsible for controlling the various hazards (i.e., enforce constraints). Note that in the Cybersafety method, each controller in the system has certain safety and security *roles* and *responsibilities* that it enforces by taking certain *control actions* based on some *feedback* about the process that is being controlled. In addition, each action or decision is taken by the controller in a given *context* (environmental conditions) in which the system is operating.

Modelling the interactions between the various controllers results in a *functional control structure* where the controllers are organized in a *hierarchy* of controllers; The behavior of lower-level controllers is controlled by higher-level controllers, which in turn are controlled by even higher-level controllers. The Capital One functional control structure is presented in Figure 3. Broadly, the controllers in the system can be abstracted into four categories: (1) Technical Controllers, (2) Operational Controllers, (3) Managerial/Leadership Controllers, and (4) Regulatory Controllers.

At the bottom of the figure the *Cloud-hosted Banking Back-end System* is shown. Importantly, this represents the cloud infrastructure that stored the sensitive credit-card application data (i.e., the controlled process that needed to be protected). One level above, a number of technical controllers are shown; these include the WAF, **Identity and Access Management (IAM)** system, Intrusion Detection System, and Configuration Management System, and so on.

Another level up, a number of Operational controllers are shown. These include the *Information Technology* team that developed and deployed the cloud programs as well as the *Information Security* team that was responsible for the security of data in the cloud. This group also includes the cloud provider—AWS—because at an operational level, it was responsible for ensuring the security of the underlying cloud infrastructure.

Further up, the figure shows higher-level controllers such as the Leadership team (including the CTO, CISO, CEO, etc.), Internal Audit, and the Board of Directors. Finally, at the highest level, the diagram shows regulatory controllers including the OCC, Federal Reserve, and so on. Note that the interactions between controllers are represented as control loops in the diagram where two types of information exchanges are expressed (1) Control Actions or decisions expressed as red arrows and (2) Feedbacks expressed as blue arrows.

Next, for each major control loop, we perform a causal analysis to understand why the control loop failed to enforce the necessary constraints on the underlying process. The causal analysis includes identifying the safety and security responsibilities for the controller, identifying the inadequate control actions and decisions, determining the context in which the incorrect decisions were provided, and, finally, identifying flaws in assumptions and process/mental models that caused the incorrect decision-making.

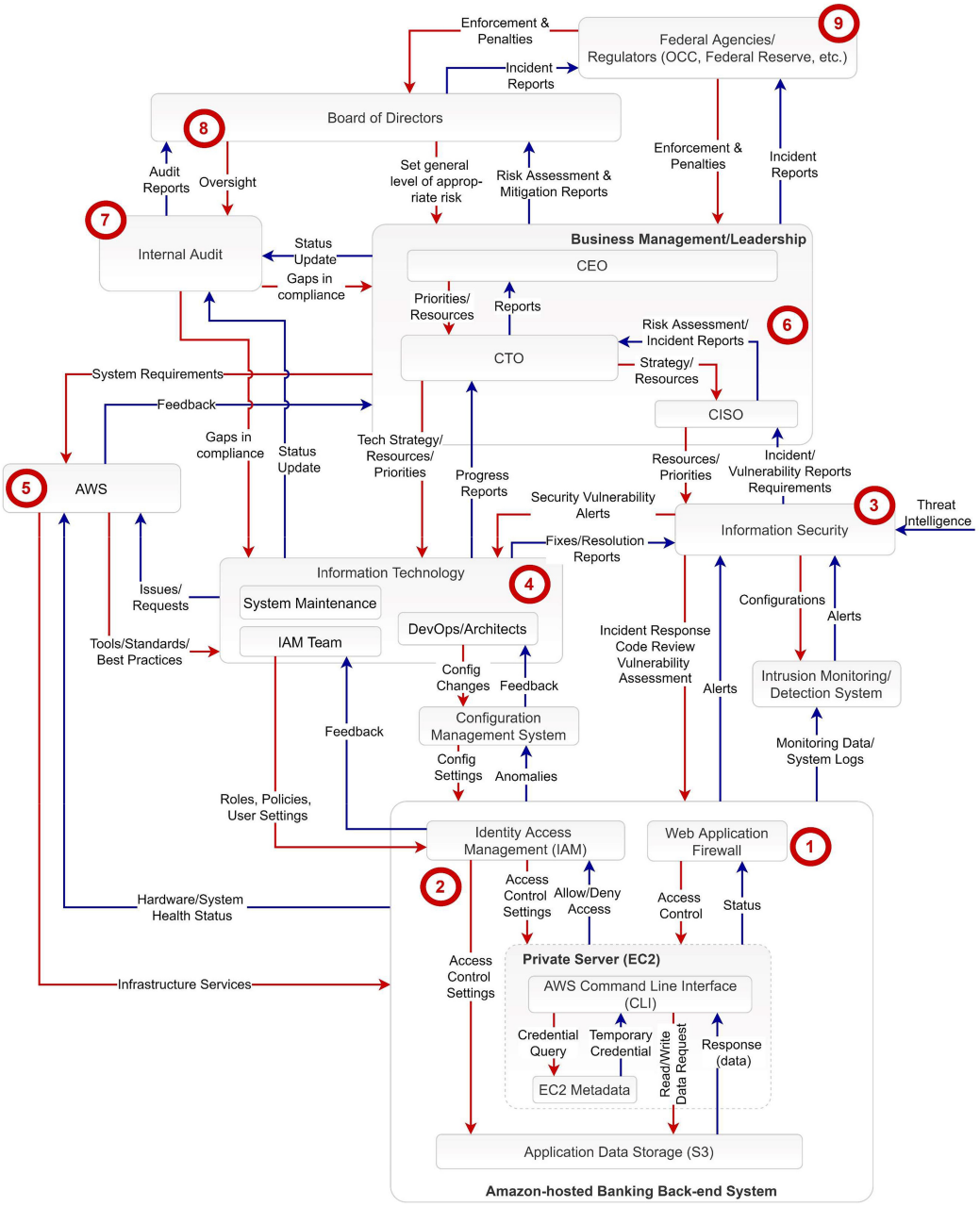

Fig. 3. Capital One functional control structure; nine key controls are noted—all of them failed.

4.2.1 *Technical Controllers.*

4.2.1.1 Web Application Firewall (Loop #1). The initial entry point in the breach was the WAF. The primary *responsibility* of the WAF was to protect a web application by filtering and monitoring HTTP traffic between the web application and the internet and protecting against common attack vectors such as cross-site forgery, **cross-site scripting (XSS)**, file inclusion, SQL injection, and

so on [33]. Similar to a network firewall that inspects and discriminates traffic based upon IP addresses and port numbers, a WAF inspects and discriminates based on HTTP traffic. A WAF can be considered to be a *reverse proxy server* as well, and in some cases it is deployed in this configuration [29]. This is because, similar to other reverse proxy servers (e.g., Nginx) that typically protect *clients* by sitting behind firewalls in private networks and direct *client* requests to the appropriate backend server, WAFs perform the same function but instead of protecting *clients* protect *servers* or specific web applications.

In the case of Capital One, the open-source *Modsecurity* [29] WAF was deployed, presumably in the reverse proxy deployment mode. However, the deployment was misconfigured, which allowed external malicious requests to reach internal resources, notably the AWS metadata service (Hazard H-3). We can identify two additional controls that were inadequate and allowed the attacker to gain initial access via the WAF. First, the **intrusion detection systems (IDS)/intrusion prevention systems (IPS)** did not raise adequate alerts about successful access from malicious IP addresses. Second, there was inadequate preventive periodic vulnerability scanning in place that could have exposed the misconfiguration.

It is important to understand the *context* in which the misconfiguration existed. On its own, the misconfiguration was not an issue per se; but when we combine this misconfiguration with the underlying AWS infrastructure and its need to use metadata service for delivering essential functionality, it becomes a critical flaw. This means that it is the lack of understanding of the interaction between the WAF reverse proxy settings and the cloud infrastructure (*process model flaw*) that ultimately resulted in the WAF exposing the backend service to outside request. This points to an issue related to understanding of roles and responsibilities under the *shared-responsibility model* between Capital One and AWS, which we address in detail in Section 4.2.2.3.

In addition, there are other indirect causal factors that potentially created the conditions necessary for the control failures described earlier. First, note that the choice of the open-source *Modsecurity* as the WAF may have been a result of Capital One's commitment to use and develop open-source technology as part of its transition to the cloud [34]. *Modsecurity* does not have a graphical user interface or central administration capability. In addition, unlike next-generation WAFs, *Modsecurity* is also unable to *automatically* detect and block phishing sites, Onion router (TOR) anonymizer addresses, botnets, and so on; a limitation that was exploited in the Capital One breach. Also, Modsecurity, if not adequately *tuned,* is known to produce a number of false positives that can result in alert fatigue—this could potentially explain why the critical alerts were not adequately dispositioned. Overall, this means that *Modsecurity* requires a significant level of configuration knowledge and continuous maintenance, which makes it error-prone.

Finally, note that the use of WAFs to protect web applications began to catch on after the release of **Payment Card Industry Data Security Standard (PCI DSS)** Requirement 6.6 [35], which necessitated automated technical solutions to protect web applications and put forth WAFs as a way to satisfy the new rule [36]. Although, PCI DSS does not apply to credit card *application data* (instead, it applies to credit card transaction data), which was exploited in the Capital One hack, we speculate that installation of the WAF may have been considered a blanket security measure for all web applications that, when applicable, would also meet the PCI DSS requirement. The deployment of the WAF, in turn, may have increased a false sense of security (*process model flaw*), since it is considered an additional layer of protection. In the process of deploying the WAF, however, it was perhaps not considered that the tool itself could become a vector in the attack.

4.2.1.2 Identity and Access Management (IAM) (Loop #2). After gaining entry, the intruder leveraged the misconfiguration to relay requests to the AWS metadata service, which returned information about the IAM role that was attached to the EC2 instance (hosting the WAF) along with temporary credentials (access token) for the role.

The primary *responsibility* of IAM is to control who is authenticated and authorized to use resources such as AWS accounts and services. For *context,* IAM is commonly used to manage *Users, Groups, IAM Access Policies,* and *Roles*. "*Roles"* is a unique IAM feature that, in essence, gives an AWS service (such as an EC2 instance) the ability to act as a user and interact with another AWS service (such as an S3 bucket). With role-based access control, there is no need to exchange security credentials or store credentials; IAM generates temporary credentials for the role (based on access policies) that expire over time.

In the case of Capital One, IAM issued temporary credentials that allowed the intruder to gain access to the sensitive S3 buckets. However, IAM functioned exactly how it was supposed (i.e., designed) to function—it received a *supposedly* legitimate request from the *trusted* AWS Metadata Service, generated temporary credentials, and returned them to the requestor.

The *process model flaw* here is that the environment inherently *trusts* the "security of the cloud" by relying on instance roles i.e., API calls coming out of a compromised instance (in this case, the AWS metadata service) are inherently trusted [6]. This is an example of a weakening of the *trust but verify* security principle in cloud architectures in the shadow of the *shared-responsibility model* (discussed in detail later) where it is mistakenly believed that the cloud service provider is responsible for verifying and preventing such unlawful access. In addition, this *process model flaw* underscores another incorrect assumption—the *belief* that newer cloud native controls and technology (that are still evolving) are better than traditional proven and mature security technologies [6].

4.2.2 Operational Controllers.

4.2.2.1 Information Technology (Loop #4). In the functional control structure (Figure 3), the **Information Technology (IT)** team is shown separate from the InfoSec/cybersecurity team; the IT team includes functions such as the system architect, DevOps, system maintenance, and IAM teams. The *roles* and *responsibilities* of the IT team encompassed day-to-day running of IT operations, setting permissions and policies for role-based access control, onboarding and offboarding users, system maintenance, as well as designing and maintaining the overall IT infrastructure. Meanwhile, the DevOps team, under IT, was *responsible* for designing, architecting, and launching new applications in compliance with various regulatory requirements.

The DevOps team within IT failed to implement a secure architecture; a vulnerability or a misconfiguration in a single chain-link led to the failure of the entire chain. However, the IAM team also failed in assigning the correct level of permissions to the "role" associated with the EC2 instance running the WAF and did not adequately encrypt the data (leading to leakage of sensitive **personal identifiable information (PII)**). In addition, it did not effectively monitor IAM security policies and roles, leading to existence of overly permissive security groups.

For *context*, the IT/DevOps team at Capital One was not in the business of developing software until very recently. In fact, in 2011, the IT team comprised 2,500 people, only 40% of whom had engineering titles. By 2018, the number of people in IT grew from 2,500 to 9,000 professionals with 85% in engineering job families [34]. "We are a software shop. We build software," said Rob Alexander, the company's CIO in [34].

One plausible scenario that explains why a misconfiguration or vulnerable application made it to production is that the shift in focus into a tech-centric company coupled with a sudden increase in tech talent (that may not be steeped in security knowledge) may have led to an unintended "lowering of guard" on security, in preference for speed of development in the collective *process model* of the IT team. Given the size of investment in technology by Capital One, this point of view is rather simplistic, but it could explain why a vulnerable application was pushed into a production environment without undergoing sufficient application code reviews/testing by the DevOps team.

Another plausible scenario is that, given the sheer size and complexity of the cloud infrastructure, the IT team just did not have full visibility of all its data; this limitation was known to upper management but the risk was considered acceptable. According to Rob Fry [37] (CTO at Jask—an Austin-based security automation company), *"With most companies, if you're be able to manage 90 percent of what you have [on your network], you're pretty sophisticated… Even if you could say you have 50,000 nodes inside AWS, and you are [monitoring] at 99 percent—which is unheard of—you still have…*(500)*… things out there that might slip through the cracks."* Note that the official announcement by Capital One about the breach [12] stated that out of the 100 million affected customers in the US, **Social Security numbers (SSN)** for about 1% were exposed and out of the 6 million Canadian customers, 1 million (about 16%) social insurance numbers were exposed; the rest were protected due to application of a *post-compromise protection measure*[7] [38] known as *tokenization*.[8]

The fact that Capital One employed post-compromise protections (in the form of tokenization) indicates that the company understood that it would not be able to watch every piece of data that resided on the cloud [37] and needed the additional layers of defense. But this raises another interesting question—why was not such an approach used to protect all PII? Some *contextual* information may provide some insights; according to Capital One [12], the compromised data consisted of application data for *secured* credit cards. Note that *secured* credit cards are typically issued to individuals who do not have established credit histories—mostly new credit cardholders, immigrants, students, or individuals with poor credit histories. Could the financial history/background of this customer population have played a role in the risk assessment of the security measures for its data? We can only speculate, but it is possible that the security of this particular application was not considered a high priority because of the content of the data it represented.

On a separate note, the functional control diagram (Figure 3) shows that the IT team received tools, guidelines, and best practices from the cloud service provider (AWS). Under the *shared-responsibility model* (described in more detail in Section 4.2.2.3), Capital One's IT team was expected to build its applications and ensure their security atop infrastructure and tools that it did not control (i.e., it was responsible for security in the cloud) while AWS was responsible for "security of the cloud," i.e., the underlying infrastructure. Here, we discuss three *process/mental model flaws* that could have potentially emanated as a result of this model.

First, overreliance on AWS best practices and guidelines could have led to a flawed decision-making process. Since the IT team did not control the underlying infrastructure, they would have *believed* that AWS best practices and guidelines were complete and following them would guarantee security. Evan Johnson (manager of product security team at Cloudflare) states [19], *"There's a lot of specialized knowledge that comes with operating a service within AWS, and to someone without specialized knowledge of AWS, [SSRF attacks are] not something that would show up on any critical configuration guide."* This indicates that following the AWS best practices and guidelines without specialized security knowledge could in fact have led to the vulnerability remaining unaddressed.

Second, from a functional control structure point of view, the overreliance on AWS best practices by the IT/DevOps team points to an additional weakness within the control structure—insufficient coordination and communication between the information security team and the development team. This indicates that rather than following the "secure by design" principle and "baked-in security," the IT/development team followed a "bolted-on security" approach, relying on multiple layers of defense, but not considering the interactions between those layers. The flawed belief here is that at least one of the layers would be able to stop the breach. For simple systems, this is

[7]Post-compromise security refers to protection of user data after the encryption key has been compromised.
[8]Tokenization refers to the process of substituting a sensitive data element with a non-sensitive equivalent, referred to as a token, that has no extrinsic or exploitable meaning or value.

an extremely valuable approach; however, as the complexity of the system increases, this linear-thinking approach becomes inadequate because of the existence of many indirect interactions between components. As we mentioned earlier, the misconfiguration itself may not have been so damaging, had the underlying cloud infrastructure not relied on metadata service for accessing roles and credentials.

Third, there may have been an incorrect belief in the principle of "*security by obscurity,*" i.e., that the environment was so complex that it was unlikely to be breached by an attacker without insider specialized knowledge. In the case of the Capital One breach, the attacker, having previously worked at AWS, indeed was knowledgeable about configuration vulnerabilities that organizations were routinely subject to, which she targeted.

Finally, to understand why the IAM roles were granted permissions wider than necessary, we highlight some contextual factors in which the IAM team operated. As Otto [37] points out, identity management is very hard to get right (quoting a cloud security engineer*), "Sometimes, the rules for these things span into six, eight pages of dense JSON text. You can't just point to a folder and say 'Administrators can read this, analysts can read that,' It doesn't work like that. It's all these weird inherited side effects. It's not that obvious at all."* Couple this implementation complexity with the fact that the installation of the WAF was required by regulation (as a means to comply with PCI-DSS Requirement 6.6—published in May 2018) with a tech-centric team focused on innovation rather than security, we can see the conditions necessary for the breach. As Mateaki [39] states, "*one of the cons of deploying a WAF is that it presents another 'bump in the wire,' and some networking re-configurations, depending on the mode of deployment.*" This potentially explains why the IAM role associated with the WAF instance was set permissions wider than necessary; it was perhaps implemented in this fashion to ensure the web application functioned as required without unnecessarily getting blocked by the firewall.

4.2.2.2 Information Security (Loop #3). The InfoSec team took the brunt of the blame for the Capital One breach; but it is important to understand why the InfoSec team failed to enforce the necessary constraints. From a *roles* and *responsibilities* perspective, the InfoSec team was ultimately responsible for data security, including vulnerability management, application security, configuration management, change management, and incidence response. It was also responsible for defining security policies for IT systems and prioritizing and alerting **Information Technology (IT)** team and senior management about critical vulnerabilities in the system as well as resource requirements.

The InfoSec team failed on a number of counts in fulfilling its *roles* and *responsibilities*. For one, the InfoSec team failed to enforce periodic preventive vulnerability scanning that would have highlighted the vulnerability. Second, the InfoSec team had inadequate detection and monitoring in place; they did not know that an intruder lurked in their network for over 127 days and exfiltrated large amounts of data. Once the breach was reported by a *white hat* hacker, they were able to *confirm* the breach and network intrusion fairly quickly. This raises the question, "*why the InfoSec team was not monitoring the network access logs in the first place*?"

These inadequacies can be partly explained by examining the *context* in which the InfoSec team operated. As mentioned previously, Capital One had a strong tech-centric culture even before it unveiled its cloud strategy in [40]. The *Wall Street Journal* [40] reports that "*Technology employees had at times been given free rein to write in many coding languages—so many that it made it harder for the cybersecurity unit to spot problems."* When Capital One began transitioning to the cloud in 2015, it suffered from a skills and knowledge gap owing to the different security requirements for operating in the cloud. This skills gap was further exacerbated by a high turnover rate, which has partly been blamed on a change in leadership in [40].

According to the *Wall Street Journal* report [40], a third of the employees in the cybersecurity team left in 2018. Although Capital One reported significant investments in cybersecurity, there is evidence that lends credence to the existence of a high stress/workload, understaffed, and high-turnover environment. For instance, in late 2017, the company bought a software called Endgame to improve its ability to detect a breach, but even after a year, the company had not finished installing the software—indicating high workload issues [40]. High turnover coupled with Capital One's commitment to developing and using open-source technology [34] (which can at times be more difficult to configure and requires more in-depth knowledge of the underlying process) may have created an error-prone environment.

We can also speculate on one *process model flaw* that might have prevented the InfoSec team from enforcing the requisite constraints—there may have been a *belief* that the layered defense approach was resilient enough to prevent a large-scale breach. After all, Capital One was following AWS best practices—a WAF was deployed, IAM was used for authentication of users and instances, and the data was encrypted. The problem is that, as was demonstrated by the breach, due to the complexity of the environment, a checkbox approach was simply not sufficient. Individually, each of the components behaved exactly as they were designed to do, but as a system, failed miserably.

Given this *context* and the *process model flaws,* it is easy to understand why the WAF was left misconfigured and why the network access logs were not being actively monitored.

4.2.2.3 Cloud Provider – AWS (Loop #5). In the immediate aftermath of the Capital One breach, the cloud service provider hosting the Capital One back-end systems, AWS, unequivocally and unapologetically denied any culpability in the hack stating that, *"AWS was not compromised in any way and functioned as designed... As Capital One explained clearly in its disclosure, this type of vulnerability is not specific to the cloud"* [41]. Let us analyze the AWS control loop to investigate this position further using our Cybersafety approach.

Note that AWS, similar to other major cloud service providers (including Google and Microsoft), offers cloud infrastructure under the *shared-responsibility model.* Under this model, the cloud service provider is responsible for the security of the cloud infrastructure, but the client (Capital One) is responsible for security in the cloud. AWS notes [42], *"Security and compliance is a shared responsibility between AWS and the customer… This shared model can help relieve customer's operational burden as AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates… The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided security group firewall."* In addition to providing secure and reliable infrastructure services, AWS is responsible for providing various tools and services along with sufficient and effective guidance to enable its customers to meet their security and compliance goals.

While it is true that in the Capital One breach the underlying infrastructure was *not* compromised, weaknesses within the design of the infrastructure were leveraged by the attacker. For instance, by design, the AWS metadata service (IMDSv1) does not verify the legitimacy of an API-request coming from an EC2 instance. The *process model flaw* here is that there is *assumed trust* in the security of the cloud—a call originating from a *compromised* instance is inherently trusted.

AWS failed on two counts with respect to this weakness—both of which have been largely dismissed by the company [26]. First, AWS did not implement countermeasures in the metadata service against SSRF and open proxy attacks that leverage this weakness despite knowing about these vulnerabilities, since at least 2018 (and despite its competitors offering some level of security against these vulnerabilities) [43]. In fact, AWS updated its metadata access service four months after the Capital One breach, taking countermeasures against these vulnerabilities. In November

2019, AWS announced that the updated IMDSv2 service provides *"defense in depth against unauthorized metadata access"* [44]. This raises the question, why was this update not rolled out sooner by AWS?

Second, AWS did not provide sufficient guidance to protect against these vulnerabilities to its customers, disclaiming any responsibility under the *shared-responsibility model*. Evan Johnson [19] notes, *"The impact of SSRF is being worsened by the offering of public clouds, and the major players like AWS are not doing anything to fix it. The problem is common and well-known, but hard to prevent and does not have any mitigations built into the AWS platform."*

To understand why AWS did not prioritize updating the metadata service, let us analyze the *environment* and *context* in which AWS operates. The enterprise cloud market is a hotly contested environment where tech giants including Amazon, Google, and Microsoft compete for market share. Although AWS currently has the largest market share at 33%, it is trailed by Microsoft and Google, respectively, which are aggressively trying to increase their footprint. As a result of this competition, there is a constant push to invent and innovate to release new services and features.

When AWS began offering its cloud infrastructure platform in 2006, it only released a handful of services; but each year, the pace of releasing new services and features has increased exponentially. AWS confirms that, *"We've also innovated and delivered at a very rapid pace (delivering 159 significant features and services in 2012, 280 in 2013, 516 in 2014, 722 in 2015, and 1,017 in 2016)..."* [44]. It further added 1,430 new services or major new features in 2017 and 1,957 in [45].

This rapid expansion in services and features has had at least three obvious, unintended detrimental effects. First, this has led to an increase in complexity of cloud services. As Stephen Harris aptly summarized in his blog, *"I really am not a fan of the AWS security model; there's far far too many knobs and controls, and it's not clear how they interact with each other. It can be hard to even know something simple ('Is this server port 22 open to the internet?') because of how configurations interact (security groups, routing tables, network ACLs, etc. etc.)"* [46].

Second, this has exacerbated the skills and knowledge gap about AWS services and features. AWS acknowledges this; Joe Chung, an Enterprise Strategist at AWS, notes [46], *"that developers can wake up to approximately 3 new changes on the AWS platform each and every day"* and that *"some organizations are asking for a series of best practices and processes that will help them keep up with the rapid pace of change."* This indicates that the rapid upgrades, although well-intentioned, have created a condition where AWS customers are lagging behind in their knowledge of the platform, tools, and services.

Third, this rapid increase in features and services unintentionally weakens adherence to the defined roles within the *shared-responsibility model*. Krishnan opines, *"... the pace at which AWS releases features, it is so easy to get caught up with the catchy names—Greengrass, Lambda, Control Tower—and delve into them without remembering the 'of' the cloud and 'in' the cloud responsibility distinction. And that oversight can prove to be very costly."* In other words, in trying to adopt the latest tools and services that the cloud provider offers, developers sometimes forget that it is ultimately their responsibility to ensure security of their applications in the cloud—not the cloud-provider's responsibility who is providing the tools and services.

Given this *context*, we can speculate two *process model flaws* that would explain why AWS did not prioritize applying countermeasures to its metadata service. First, AWS may have believed that to increase market share it had to innovate and add new features and services; customers demand this over fixing of prior known issues. Joe Chung's blog [46] hints about this thinking when he writes about the reason for customers demanding best practices from AWS, *"Their goal, of course, is to embrace and take full advantage of what's new and improved at AWS."* The *flaw* over here is that perhaps senior leadership at AWS did not have the requisite feedback from security personnel at

the partner companies; the flow of information was limited to that between developers and AWS and hence the focus was on innovation and speed rather than security.

Second, the *shared-responsibility model* may have led to the belief that the guidelines and security best practices were adequate; even if they were inadequate, in case there was a security incident, it was the customer's responsibility to ensure their applications were secure and to ensure that their staff was adequately skilled and trained. These potential *process/mental model flaws* in the *context* of the competitive environment that AWS operated in explain why AWS may not have prioritized updating the metadata service to counter the known vulnerabilities.

4.2.3 Management Controllers.

4.2.3.1 Senior Leadership (Loop #6). Although the Capital One breach was ultimately blamed on a *"misconfigured firewall,"* many of the policies and decisions taken by senior leadership, years before, created the conditions necessary for the breach. Insofar as cybersecurity is concerned, the primary *responsibility* of senior leadership was to provide a meaningful cybersecurity strategy and allocate sufficient resources to achieve their security goals in their transition to the cloud based on adequate risk assessment and management processes.

Note that Capital One was the first large bank to adopt the public-cloud infrastructure. This strategy was at odds with the rest of the financial industry, which was foraying into risk-averse private cloud approaches (which was also favored by the regulators at the time) [47]. This is because the industry, in general, was apprehensive about the security and multitenancy aspects of the public cloud. Since Capital One had previously attempted a private-cloud approach (and failed), the leadership believed that to remain competitive, they needed to rely on the public cloud and selected AWS as the sole service provider [47]. This decision, in turn may have been influenced by the acquired tech-talent that demanded this. Rob Alexander, the company's CTO said in 2018, *"We picked AWS because they are the leader in the market. We picked them because our developers demanded it. We picked them because of the features and capabilities that support large enterprises"* [34].

Capital One's [48, 49] annual reports provide evidence that the company continuously acknowledged its increased risk exposure due to the outsourcing of substantial amount of infrastructure to AWS. The company also realized that it may specifically face an increasing number of cyberattacks as it expands its used of cloud technologies (p.24 [48]). This validates the assumption that the management was in fact aware of the increased risks associated with migration to the cloud (p.23 [48], p.40 [49]).

However, the senior leadership failed to establish adequate risk management processes for its cloud strategy, "including appropriate design and implementation of network security controls, adequate data loss prevention controls, and effective intrusion detection and monitoring controls" [13].

One of the reasons for this inadequate action could be a *flawed process model.* Note that the functional control diagram (Figure 3) shows interactions between senior leadership and AWS. Given that the senior leadership understood the increased risks of migrating to the cloud, the flawed process model could be a result of a misunderstanding of the cloud *shared responsibility model.* This potentially led to a blind *trust and pure reliance* on the cloud provider's security model. The assumption being that AWS was much better equipped to ensure security of the infrastructure as well as respond to security incidents than Capital One itself (in the cloud as well as on premise). Therefore, Capital One could focus on the rest of the stack instead, i.e., focus on developing applications to enhance user experience. In fact, one month before the breach, the company's CEO, Richard Fairbank, said, *"A lot of how we built our company is not by studying banking, but by forgetting about banking"* [50]; indicating an *inherent trust* in the security of the underlying cloud infrastructure.

This *inherent trust* in the cloud provider's security model potentially also led to an elevated risk appetite stemming from use of new technologies. The fact that the company enhanced its cybersecurity governance after the incident lends credence to this hypothesis. The 2019 annual report [49] notes that the leadership established a senior management level committee mandated to support reporting of cybersecurity-related issues to the Board to fulfill OCC's consent order.

In addition to understanding the flaws in the process model, we have to understand the *context* in which the decisions were made. Prior to unveiling its cloud strategy in 2015, Capital One's leadership strategized that to remain competitive they had to reinvent the bank into a forward-looking technology company. This strategy included reducing technology (data center) costs while taking advantage of the cloud's rapid scalability and ever-increasing array of applications. The success of this strategy, however, was predicated on the bank's ability to aggressively attract and acquire tech talent. For this, the company's leadership committed to some bold decisions including *open-source* technology, *agile* development philosophy, and transition to the public cloud.

Note that we are not implying that the strategic decisions taken by senior leadership were wrong. The company made critical steps toward establishing an appropriate risk management regime. For example, in 2018, the company appointed the Chief Operational Risk Officer responsible for establishing and overseeing the company's operational risk management program, which included topics such as internal and external fraud, cyber and technology risk, data management, model risk, third-party management, and business continuity (p.73 [49]). Instead, we argue that the pace of maturity of the cyber risk management practices was inconsistent with the pace of adoption of new technologies and the acquisition of risks associated with using them. In addition, we highlight the interdependence of policies at higher levels of the organization with operational decisions at the lower levels. For instance, the choice of the firewall, considered the initial target of the attack—the open-source Modsecurity WAF—can be traced back to the commitment by senior management to open-source technology.

Likewise, the commitment to public cloud (in direct contradiction to industry practice) in conjunction with acquisition of tech talent and blatant push by senior management to *"not think like a bank"* inadvertently increased the appetite for riskier approaches. The emergent consequence of this strategy (agile, open-source, and public cloud) was an unintended shift in priorities to speed and cost to develop new applications at the expense of security of those applications, which is clearly evident in the multiple control failures of the Capital One breach.

In the next section of the article, we would like to assess the reasons that influenced the decisions of the company's board regarding cyber risk management.

4.2.4 Board of Directors and Internal Audit (Loop #7, #8). The board of directors and internal audit play a key role in the oversight of company's cyber risk management. The Capital One annual reports of [48, 49] confirmed that its board assumed the responsibility of cyber risk oversight, and it had an established risk committee that oversaw the company's cyber risk profile, top cyber risks, enterprise cyber program, and key enterprise cyber initiatives.

We assessed the cyber risk management practices that the board employed according to the company's annual reports [48, 49] against the principles proposed in cyber-risk oversight handbook by the National Association of Corporate Directors [38]. The summary of the analysis is shown in Table 2.

It is evident from the analysis that the company's board of directors had technically employed all five principles before the incident to comply with regulatory requirements; only few adjustments were made after the breach. This raises two interesting questions: First, why did so many failures occur in Capital One with such a strong board? Second, what is the reason why meeting compliance requirements does not effectively prevent security incidents? From the functional

Table 2. Summary of Capital One's Cyber Risk Management Practices before and after the Breach

#	Principle	Capital One 2018–2019 (pre-breach)	Change after breach
1	Directors need to approach cybersecurity as an enterprise-wide risk issue.	Capital One is a digital bank and technology and cybersecurity played a critical in the company's business. All board directors had experience in digital, technology, and cybersecurity and should have realized the integral nature of cyber risks.	OCC demanded to develop risk assessment processes to identify and manage technology risks within the cloud operating environment.
2	Directors should understand the legal implications of cyber risk.	7 out of 10 board directors had experience with regulated businesses, regulatory requirements, and relationships with state and federal agencies.	No major changes for the board members. Only one member without specific technology or cyber security expertise was added to the board after the incident.
3	Boards should have access to cybersecurity expertise and allocate sufficient time to discussions about cyber risk management on a regular basis.	**Expertise** All board members had significant cybersecurity and technology experience expertise. One of the board directors was a former Amazon CISO. The Risk Committee met third-party experts to evaluate the company's enterprise cyber program. **Allocation of time to cybersecurity** The Risk Committee receives quarterly reports from CISO on the company's cyber risk profile and enterprise cyber program and meets CISO at least twice annually.	No changes on the board members to obtain additional cybersecurity expertise. Meetings with CISO became quarterly.
4	Directors should demand establishing an enterprise-wide cyber risk management framework with adequate staffing and budget.	The board reviews and discusses the company's technology strategy with CIO and approves the company's technology strategic plan at least annually. The Risk Committee oversees cyber, information security, and technology risk, as well as management's actions to identify, assess, mitigate, and remediate material issues.	Enhanced Board Oversight of Cybersecurity Risk. The Board's engagement on cybersecurity has been heightened and the Board is overseeing multiple enhancements management is making to Capital One's cybersecurity standards, policies, procedures, and processes. This effort is intended to strengthen Capital One's cybersecurity risk management capabilities, including reporting on these risks to the Board and its Committees. The Risk Committee has taken the lead in overseeing this effort, and the full Board has also been actively engaged.
5	Board-management discussion about cyber risk should include identification of which risks to avoid, which to accept, and which to mitigate or transfer through insurance.	The company had a $400M cyber insurance policy [51]. The Risk Committee annually reviews and recommends the Company's information security policy and information security program to the Board for approval. The Risk Committee receives updates from management on its cyber event preparedness efforts and reviews reports from CIO and CISO on significant cyber incidents.	CISO was included into regularly meetings with the board.

control diagram (Figure 3), one potential *process model flaw* is obvious; the board did not have a direct communication channel (feedback) with the CISO to receive information about potential risks from the company's main security defender. In *Cybersafety* terminology, this would have resulted in the board forming an inadequate process model of the bank's risk exposure, leading to ineffective control. Interestingly, the lack of presence of CISO at the executive level has remained a common theme in the majority of companies that have experienced a significant breach [4, 11].

The requirements imposed by the OCC consent order [13] enable us to further understand the nature of the deficiencies at the board level. First, the OCC demanded a development of appropriate and effective risk assessment processes to identify and manage technology risks within the cloud operating environment (including risk assessment processes specific to technology changes) [13]. This requirement makes it clear that the board probably missed to prioritize the importance of the cyber risk management related to the new technology-related risks, including cloud-related security risks, despite acknowledging their existence. We suppose that the weak feedback loop mentioned above was the major cause of it.

The second demand was to reassess the quality and transparency of cyber and technology risk reporting at the board level. Since the company included the CISO in the meetings with the board as a reaction to the cyber breach, we may conclude that prior to the breach, cyber risks were not comprehensively covered in the communications received by the board. They could have been potentially limited to providing formal reports to meet compliance requirements, but not revealing actual risks. These two facts have led us to believe that despite establishing cyber risk management practices, the maturity of these practices was insufficient to identify and address risks emerging from the adoption of new technologies. This was an unexpected finding, given that one of the board members was the former CISO of Amazon—a leader in cloud computing business.

Another two OCC requirements demanded the company's board to increase scrutiny, monitoring, and oversight of management's actions to address (internal) audit findings as well as hold management accountable for the timely remediation of the identified material risk issues [13].

These requirements have highlighted another deficiency in the company's risk management practice—internal audit. OCC identified that internal audit failed to identify numerous control weaknesses in the cloud operating environment and did not report on the identified weaknesses and gaps to the board audit committee. Moreover, some concerns raised by internal audit did not find an appropriate reaction from the board that failed to hold management accountable for the resolution of the issues [13]. We have not been able to find any publicly available information on the probable cause of it at Capital One. However, a recent research study shows that advanced and enabling technologies (such as cloud technology) is the primary area where 38%–49% companies are looking to enhance their internal audit skills [52]. Therefore, we can hypothesize that the lack of expertise in the cloud technologies could have been one of the reasons why internal audit failed to identify the risk and the control weakness.

In summary, we can conclude that the formal adherence to the principles of cyber risk management at the board level itself is not sufficient to effectively prevent a data breach.

4.2.5 Regulatory Controllers (Loop #9). In the US, a broad range of federal and state agencies, including the OCC, the **Federal Deposit Insurance Corporation (FDIC)**, the **Office of Thrift Supervision (OTS)**, and the banking departments of various states, regulate financial institutions. The OCC charters, regulates, and supervises nationally chartered banks, while the FDIC, the Federal Reserve, and state banking authorities regulate state-chartered banks. The Federal Reserve regulates bank holding companies and financial services holding companies, which own or have controlling interest in one or more banks. Meanwhile, the OTS examines federal and many state-chartered thrift institutions, which include savings banks and savings and loan associations. The

goal is to enforce the establishment of robust information security practices by the financial sector to protect consumers and ensure the stability of the economy [53].

At the federal level, Capital One is subject to Section 501(b) of the **Gramm-Leach-Bliley Act (GLB Act)** and Section 216 of the **Fair and Accurate Credit Transactions Act of 2003 (FACT Act)**. In pursuance of these acts, the Board of the Federal Reserve System along with other federal banking agencies (including the OCC, FDIC, OTS, and Department of Treasury) issued *Interagency Guidelines Establishing Standards for Safeguarding Customer Information* [54]. The guidelines establish standards relating to administrative, technical, and physical safeguards to ensure the security, confidentiality, integrity, and the proper disposal of customer information. Financial institutions are expected to follow the guidelines and test the key controls, systems, and procedures of its information security program. From a *roles* and *responsibilities* perspective, for Capital One, the OCC has the authority to enforce compliance with the guidelines.

In addition to the federal guidelines, some states also impose further regulations on financial institutions. For example, effective March 1, 2017, New York's Department of Financial Services had promulgated Cybersecurity Requirements for Financial Services Companies (Requirements) operating in New York, which went into full effect in 2019. Since Capital One operates in New York, it is subject to these Requirements as well.

Under these Requirements, Capital One is tasked to protect information systems and nonpublic information from cyber-threats by developing and implementing a comprehensive and effective cybersecurity program. The Requirements also require senior management to take cybersecurity risks seriously, be responsible for the organization's cybersecurity program, and file an annual certification confirming compliance with the Requirements. The Requirements specify expectations across all major areas of a cybersecurity program, including risk assessment, penetration testing, audit trails, limit access privileges, a mandatory CISO reporting annually to the board, encryption, multi-factor authentication, and vendor compliance.

As a result of these regulations, it was noted that the board was much more attentive to cybersecurity in the case of Capital One compared to Equifax. However, despite the board's attention to cybersecurity and the bank's apparent compliance with all the requirements, Capital One suffered a major breach indicating the inefficiency/ineffectiveness of the regulations. Given that Capital One technically satisfied all the requirements, it can be inferred that the compliance to the guidelines and requirements was treated as a *checkbox exercise* (i.e., bureaucratic demonstration of adherence to rules without actually understanding the implications that would improve security), which was just not sufficient to ensure cybersecurity. In their proposed research plan to understand the role of compliance in improving or hindering cybersecurity, Marotta & Madnick [55] note that organizations often "tick" the compliance "checkbox" and do the bare minimum just to pass regulatory audits. They further note that this is a common approach that *"prevents organizations from being able to reflect on the impact that regulations may have on security"* [55] implying that regulatory compliance is not necessarily a measure of effectiveness of regulations due to flaws in implementation.

Furthermore, understanding the *context* in which the various regulators operate and the *web of regulations* that a financial institution, such as Capital One, would have to navigate to demonstrate regulatory compliance, it can be noted that the majority of regulators demand similar cybersecurity requirements and practices but use differing vocabularies and frameworks to do so. These requirements are continuously layered on institutions creating unnecessary complexities and drawing off substantial personnel and resources from cybersecurity to the reconciliation of the various agencies' cybersecurity expectations and proper reporting to examination requests. It was estimated that up to 40% of the security teams' time can be consumed by these activities [56].

The Capital One breach provides evidence that excessive regulations and multi-agency enforcement efforts do not provide effective consumer protection against data breaches, but can hinder it. This is further evidenced by the fact that a financial institution spends three times as much as a comparably sized non-financial institution on security [57]. The recent major cyber incidents with Capital One and Equifax have unearthed serious questions about the efficiency of the cybersecurity compliance regime. Does the growing number and complexity of regulations proportionally impact the cybersecurity of financial institutions?

The Cybersafety analysis of the Capital One breach sheds light on another interesting angle with respect to regulators, i.e., lack of regulatory oversight for cloud service providers. The leading cloud providers such as Amazon, Microsoft, and Google have become a critical part of the IT infrastructure of almost all modern companies—89% of all Fortune 500 organizations confirmed using one or more of these three vendors in 2019. Financial services institutions specifically have also increased reliance on cloud service providers, thus creating new risks to manage.

We learned from the Capital One incident that one of the major risks with respect to cloud service providers stems from the misunderstanding of the *shared security responsibility model*. The proper understanding of the division of responsibilities for assessing and implementing appropriate controls is critical especially because cloud providers are not subject to any specific cybersecurity regulations and all cyber-related risks with third parties are expected to be managed by the financial institution. Although, the Capital One case demonstrated how AWS tried to completely deny all culpability for the weak cyber defense, this position will hardly be acceptable to the courts. The class action lawsuit filed in August 2019 against both Capital One and AWS claimed that neither Capital One nor AWS adequately safeguarded data making them vulnerable to hackers. This confirms that the call for accountability of cloud service providers will gain more traction in the future.

5 RECOMMENDATIONS

In the previous section, we analyzed the functional safety control structure for Capital One to understand the causal factors for the breach; we uncovered several process model flaws, inadequate control actions, and decisions as well as contextual factors that created the conditions necessary for the breach. Based on these findings, in this section, we outline some recommendations that would help improve the security of the enterprise by taking a systems perspective of the security problem. These recommendations do not span only technical controls, but also operational, organizational, as well as regulatory controls, and as such apply not only to Capital One, but any enterprise that is serious about improving its security posture.

Technical Controllers. In the Capital One case, there were multiple technical lapses in the configuration and programming that led to the success of the breach. Our first recommendation is to emphasize the need for a robust application review/vulnerability assessment process to be periodically instituted to ensure vulnerable applications are identified and adequately patched. In the case of Capital One, a misconfigured reverse proxy remained undiscovered. With attacks such as SSRF, reverse proxy, and so on, taking center stage, there is no excuse for organizations to not prioritize detection and patching of such vulnerabilities.

Another key finding was that even on the technical level, there was an implicit reliance on the security of the cloud. It is important to understand the roles and responsibilities under the shared responsibility model and revisit the *trust but verify* security principle. In the case of Capital One, the AWS metadata service inherently trusted an API call from an EC2 instance without verifying its malicious origin. Although AWS has issued a fix for this vulnerability in version 2 of its metadata service, we believe it is important to emphasize this point as a general principle when developing cloud services.

Third, the least privilege security principle must be enforced for all IAM policies and roles (despite the complexity of managing multiple IAM roles and policies). In Capital One's case, the IAM permissions associated with the WAF role were excessive; S3-related permissions were provided to the IAM role associated with the WAF that led to the breach. According to Walikar [5], *"a lot of developers and Ops, to make things work, still rely on the dangerous 'Effect': 'Allow,' 'Action': '*,' 'Resource': '*'[9] policy effectively giving the IAM Role AWS Administrative rights."* From the functional control structure's perspective, this means the IAM team must develop processes to ensure effective monitoring of IAM security policies and roles.

Operational Controllers. Next, we transition to recommendations for operational controllers. In the functional control structure, from the InfoSec team control loop, we noted that the Intrusion Monitoring/Detection system was either missing, did not raise the necessary alarms, or was ignored by the InfoSec team that enabled the breach. The first recommendation that stems from the Cybersafety analysis is the implementation of a platform that effectively monitors IAM and AWS **Security Token Service (STS)** API calls and raises alarms for any anomalous behavior, such as API calls originating from the WAF instance resulting in reading or writing of data in S3 buckets. Having an effective Intrusion Monitoring/Detection system would raise the necessary alerts and enable the InfoSec teams to take the necessary control actions.

Second, the operational controllers (DevOps, Security Architects, etc.) must also adhere to the *trust but verify* security principle in architecting cloud services. It is critical to understand the roles and responsibilities of each party under *shared-responsibility model.* While developers and architects should use the tools, guidelines, and best practices provided by the cloud provider, they must understand their responsibility to ensure security of applications in the cloud by *verifying* the adequacy of the deployed tools and technologies. It is also extremely important for the operational controllers to approach security as a system issue. In the Capital One breach, the control failures (from the misconfiguration, to the interaction of the reverse proxy with the backend resource, to the over-provisioning of IAM roles, to lack of use adequate encryption technique, and finally lack of adequate IDS/IPS) show a lack of understanding of the system and lack of coordination and communication between disparate entities of the security and development teams. It is therefore recommended that the security function should be implemented as part of the development function with close coordination and coupling between them. This would ensure that security is *baked* into the product with an end-to-end understanding of the system.

For its part, the cloud service provider must ensure that its infrastructure and services are secure by design, their tools, guidelines, and best practices are free of any bugs, and any known vulnerabilities/misconfigurations are explicitly known by the customer. In our analysis of AWS' control loop, we noted that that there was a tremendous focus on innovation and deployment of new tools and services. While well-intentioned, such a rapid increase in growth of services inadvertently increases technical debt and causes a knowledge and skills gap with the customers. For instance, vulnerabilities such as SSRF, reverse proxy, and so on, were known to the cybersecurity community much before the Capital One breach, yet it was only several months after the breach that AWS issued an updated metadata service that provided protection against such vulnerabilities. Our recommendation is that the cloud service provider should revisit their pace of releasing new services and instead focus on the cloud architecture and ensure simplicity of design. There should also be a concerted effort to limit the complexity of tools and services to ensure security is simplified.

[9]The sample IAM policy can be translated as such: use of the wildcard (*) in the IAM JSON policy element "Action" specifies that all actions are allowed, while the use of the wildcard (*) in the "Resource" element specifies that access to all resources is allowed.

Management Controllers. As far as senior management is concerned, in the case of Capital One, we saw a strategic shift in the focus of the company from a bank to a software company in the years preceding the breach. Several bold and risky decisions were taken to propel the bank into a forward-looking tech company, including a large acquisition of tech talent to support the transition. While the strategic decisions were well-intentioned, they inadvertently shifted focus from security (i.e., a core function of the bank) to growth and innovation. The incorrect assumption was that the cloud service provider was responsible for security so the bank could focus on enhancing customer experience through innovation and new software development. This in turn potentially led to an elevated risk appetite throughout the organization, which ultimately paved the way for the breach. In light of this, our recommendation for senior management is that they must ensure that the pace of maturity of the cyber risk management practices is consistent with the pace of adoption of new technologies and the acquisition of risks associated with using them. Similar to AWS, the leadership team must resist the urge to grow too fast at the expense of security.

Second, the *shared responsibility model* should be revisited; it is critical to understand roles and responsibilities of both parties at every level of the enterprise—not just at the operational level but also at the leadership level. Rather than assuming *blind trust* in the cloud service provider's security model, the leadership team must emphasize the criticality of ensuring security to their organization through their actions, including adequate resource allocation and priority/goal-setting. Finally, the high turnover rate in the InfoSec team was a clear indication that all was not well at the operational level and required immediate action by the senior leadership. The recommendation is that the senior leadership must update its *process model* based on feedback from the operational teams and take necessary corrective action before it is too late.

At the board level, specifically, boards should probe a company's cloud-related practices, especially with regards to assessment of any enterprise-grade security systems and analytics, determination of attack vectors, and review of data security measures. Boards should also confirm that a company has comprehensive means to prevent sensitive data from being uploaded to the cloud for inappropriate sharing, along with requisite visibility and access to detect anomalies, to conduct further investigation and undertake prompt and decisive remedial action. Boards should ensure that cyber risks are comprehensively covered in the communications received by the board. Along these lines, it is recommended that the boards should receive direct feedback from the CISO.

Along these lines, questions should cover technologies used to prevent the unauthorized use of cloud applications by employees; internal controls regarding any cloud applications used by employees; an incident response plan for handling an attack on any cloud application; and employee training concerning use of cloud applications.

Regulatory Controllers. Finally, the Capital One breach revealed weaknesses in enforcement of security requirements by regulators. Although the regulators were successful in ensuring that the Capital One board of directors paid close attention to security and that the bank demonstrated compliance with all regulatory requirements, they failed to improve security. One of the reasons we postulated was the use of a *checkbox* approach that limited the effectiveness of the regulations.

Note that the *checkbox* approach is a manifestation of a shift in focus away from security to bureaucratic compliance. We believe this is a result of layering of multiple requirements by various regulatory bodies on financial institutions without coordinating and reconciling among themselves, resulting in undue overhead on the financial institutions. Therefore, the recommendation for the regulatory bodies is to coordinate among themselves and conduct a comparative analysis of the various cybersecurity regulations applicable to financial institutions, identify commonalities between them, and harmonize and simplify them. Such simplification would reduce the overhead on security resources at financial institutions, enabling them to focus their attention instead on enhancing security.

In addition, our analysis noted a lack of regulatory oversight governing cloud service providers in the shadow of the *shared-responsibility model*. Note that the Capital One breach was, in part, possible due to an underlying weakness in the cloud infrastructure (which was only rectified by AWS after the breach). Given this and other recent high-profile breaches in the cloud, we recommend a need for regulatory oversight over the cloud service providers to enhance accountability and ensure that insecure products are not pushed to the consumer under the *shared-responsibility model*.

6 CYBERSAFETY PERFORMANCE EVALUATION

In this section, we provide a subjective critical evaluation of the Cybersafety method in identifying causal factors that enabled the Capital One breach. Note that the underlying framework (set of assumptions) for the Cybersafety method is the powerful and well-established STAMP framework [10] that is based on Systems Theory. There are several examples in literature where STAMP-based methods have been used to analyze accidents [58–63]. These analyses have successfully identified systemic causal factors that encompass the larger socio-technical and socio-organizational system, shedding new light on causes for accidents. Our research has indicated that the same approach (with some adaptations) can be successfully extended to study cyber incidents as well to uncover structural and mental model flaws indicating systemic and organizational weaknesses. In prior work investigating the TJX cyberattack [11], we compared the results of the Cybersafety method with recommendations by the **Federal Trade Commission (FTC)** and discovered that the Cybersafety method captured all the recommendations by the FTC but additionally provided insights that the other investigations either missed altogether or provided incompletely. The intent of this research was to identify critical lessons that can be learned from the Capital One cyber incident by studying the organization as a whole and investigate factors beyond the *misconfigured firewall* explanation that has been popularized to reduce the cause of the cyber-incident to an accidental human-error. To that end, our analysis uncovered organizational and systemic causal factors and flaws in a structured fashion throughout the larger socio-organizational system that caused the misconfiguration to exist in the first place.

It is important to note that the Cybersafety methodology is not limited to studying special cases only but can be adequately applied to study cyber incidents of any size and complexity; the only caveat is that the success of the analysis is dependent on the availability of information. In the case of the Capital One breach, we were limited to using publicly available documents to piece the details of the incident together (sometimes in the face of conflicting information). Let us subjectively analyze the costs (time and difficulty) associated with each step of the analysis.

Step 1 (collecting necessary information and proximate events) was of moderate complexity to implement. The Capital One incident was extensively written about in the media; but the biggest challenge was navigating the conflicting information in the news reports. It was widely speculated that the breach was a result of an SSRF vulnerability, while the cloud-provider emphatically denied this speculation, arguing instead that it was a misconfiguration of the firewall (targeted through a reverse-proxy attack). Overall, this step consumed two to three weeks to scrutinize the various news reports and develop a benchmark for the incident. Step 2 of the analysis (modeling the hierarchical functional control structure) required several iterations to get right. We used publicly available organizational archetypes for large banks and financial institutions along with logical reasoning to create the hierarchical functional control structure; the basic structure, however, was agreed upon by the research team in about two weeks' time. This step could potentially take less time if there was input from the affected organization. However, it should be noted that this step requires the analyst to be proficient in modeling functional diagrams.

Step 3 (examining missing/ineffective controls) of the analysis was aided in part by prior work done by Neto et al. [7] in identifying various controls that were missing, inadequate, or ineffective

that enabled this cyberattack to be successful. However, this prior study [7] was mostly limited to technical controls. Overall, this step was straightforward and took about one to two weeks to complete. We combined this step of the analysis with Step 4 (identifying flaws in the control structure), since we were additionally interested in identifying systemic factors that resulted in the breach. We used public incident reports from Step 1 to extend our understanding of various control failures, process model flaws, and structural flaws. This part of the analysis took about three to four weeks to compile. Finally, Step 5 (create recommendations) naturally flowed from the analysis and took two to three days to complete.

Overall, each phase of the analysis highlighted different aspects of the incident, which deepened our understanding of systemic and organizational factors at play that enabled the incident to occur, shedding light beyond the oft-repeated *firewall misconfiguration* as the root-cause of the incident. While the Capital One was a significant breach with many unique aspects that made it an interesting analysis from an academic perspective, it is important to note that the methodology demonstrated in this article provides a structured, systematic, and repeatable set of steps to analyze any incident of any size and complexity.

7 CONCLUSIONS

Most organizations that suffer a major catastrophic event (accident or cyber incident) usually perform a post-accident investigation to identify the root-cause of the incident. While well-intentioned, the focus is on what went wrong rather than why it went wrong. Clouded by hindsight bias, the outcome of such an investigation is typically the identification of a lone operator that made an error in judgment or a single piece of technology that unexpectedly failed. The natural consequence is an increased focus on improving training of operators or enhancing reliability of technology, without undertaking a deeper holistic view of the problem. Rarely is the design of the system questioned; questions such as, "was the operator set up for failure due to the design of the system or was the technology improperly deployed due to incorrect assumptions?" are rarely addressed.

In our Cybersafety analysis of the Capital One data breach, we attempted to go beyond the identification of a singular *misconfigured firewall* as the root-cause of the incident. Instead, we took a holistic view of the system. What resulted was a discovery of systemic conditions, indirect interactions, flaws in assumptions and mental models, and contextual factors along with incorrect or inadequate decisions that transcended the technical layer and ultimately resulted in the loss. Our analysis deliberately attempted to take a holistic view of the system to understand not only *what* went wrong, but *why* it went wrong. The Cybersafety method described in this article can be replicated by other organizations to study other security incidents. The findings and recommendations are also universal and apply to any organization that is serious about improving its security posture.

REFERENCES

[1] B. Kammel, D. Pogkas, and M. Benhamou. 2020. These are the worst cyber attacks ever. Retrieved from https://www.bloomberg.com/graphics/corporate-hacks-cyber-attacks/.

[2] Capital One. Information on the Capital One Cyber Incident. Retrieved on 24-July-2022 https://www.capitalone.com/facts2019/.

[3] U.S. Department of Justice, US District Court for the Western District of Washington at Seattle. United States of America vs. Paige A. Thompson, a/k/a 'erratic'. Case No. MJ19-0344, Filed 07/27/19 [Online]. Retrieved on 24-July-2022 https://www.justice.gov/usao-wdwa/press-release/file/1188626/.

[4] I. Kabanov and S. Madnick. 2021. Applying the Lessons from the Equifax Cybersecurity Incident to Build a Better Defense. *MIS Quarterly Executive* 20, 2, Article 4 (2021). https://aisel.aisnet.org/misqe/vol20/iss2/4.

[5] R. Walikar. 2020. An SSRF, privileged AWS keys and the Capital One breach. Appsecco. Retrieved from https://blog.appsecco.com/an-ssrf-privileged-aws-keys-and-the-capital-one-breach-4c3c2cded3af.
[6] J. Villa. 2020. The Capital One breach: Did the technology or the process fail? Retrieved from https://www.guidepointsecurity.com/how-aws-best-practices-hurt-capital-one/.
[7] N. Novaes Neto, S. E. Madnick, A. Moraes, G. de Paula, and N. Malara Borges. 2020. A case study of the Capital One data breach. *SSRN Electron. J.* (Mar. 2020).
[8] N. Leveson. 2004. A new accident model for engineering safer systems. *Safety Science* 42, 4 (2004), 237–270. https://doi.org/10.1016/S0925-7535(03)00047-X.
[9] S. Khan and S. E. Madnick. Cybersafety: A System-theoretic Approach to Identify Cyber-vulnerabilities & Mitigation Requirements in Industrial Control Systems. *IEEE Transactions on Dependable and Secure Computing*. DOI:10.1109/TDSC.2021.3093214
[10] N. G. Leveson. 2018. *Engineering a Safer World.* The MIT Press.
[11] H. Salim and S. Madnick. 2016. Cyber Safety: A Systems Theory Approach to Managing Cyber Security Risks-Applied to TJX Cyber Attack. Retrieved on 24-July-2022 http://web.mit.edu/smadnick/www/wp/2016-09.pdf.
[12] Capital One. 2020. 2019 Capital One cyber incident. What happened. Retrieved from https://www.capitalone.com/facts2019/.
[13] United States of America Department of the Treasury, Office of the Comptroller of the Currency (OCC), "Consent Order for the Assessment of a Civil Money Penalty", Case No. AA-EC-20-51. Retrievd on 24-July-2022 https://www.occ.gov/static/enforcement-actions/ea2020-036.pdf.
[14] A. Andriotis. 2020. Capital One senior security officer being moved to new role. *Wall Street Journal.* Retrieved from https://www.wsj.com/articles/capital-one-senior-security-officer-being-moved-to-new-role-11573144068.
[15] J. Shukla. 2020. Understanding the Capital One attack. When WAFs Fail. Retrieved from https://www.k2io.com/understanding-capital-one-attack/.
[16] G. Steel. 2020. The Capital One breach and cloud encryption. Cryptosense. Retrieved from https://cryptosense.com/blog/the-capital-one-breach-and-cloud-encryption/.
[17] J. Stella. 2020. A technical analysis of the Capital One cloud misconfiguration breach. Retrieved from https://www.fugue.co/blog/a-technical-analysis-of-the-capital-one-cloud-misconfiguration-breach.
[18] C. Morrison. 2020. The technical side of the capitol one AWS security breach. Retrieved from https://start.jcolemorrison.com/the-technical-side-of-the-capital-one-aws-security-breach/.
[19] B. Krebs. 2020. What we can learn from the Capital One hack — Krebs on security. Retrieved from https://krebsonsecurity.com/2019/08/what-we-can-learn-from-the-capital-one-hack/#more-48424.
[20] E. Flitter and K. Weise. 2020. Capital One data breach compromises data of over 100 million. *The New York Times.* Retrieved from https://www.nytimes.com/2019/07/29/business/capital-one-data-breach-hacked.html.
[21] AWS. 2021. Capital One enterprise case study—Amazon web services (AWS). Retrieved from https://aws.amazon.com/solutions/case-studies/capital-one-enterprise/.
[22] US Attorney's Office. 2021. Bank employee charged with stealing more than $100K from customer accounts. USAO-CT. Department of Justice. Retrieved from https://www.justice.gov/usao-ct/pr/bank-employee-charged-stealing-more-100k-customer-accounts..
[23] Capital One. 2021. Capital One reports inside job data breach. Retrieved from https://oag.ca.gov/system/files/CA AG Notice Remediation Letter 41952117_0.pdf.
[24] C. Folini. 2020. "Christian Folini on Twitter:".@briankrebs explains the @CapitalOne breach as a Server-Side Request Forgery due to a misconfigured #ModSecurity. I'm intrigued, but I can't see how you could configure the #WAF for #SSRF – and I wrote the #ModSecurity hand. Retrieved from https://mobile.twitter.com/ChrFolini/status/1157533808402620416.
[25] Alberto Wilson and G. Gabarrin. 2022. SSRF's up! Real world server-side request forgery (SSRF) Shorebreak Security—experts in information security testing. Retrieved from https://www.shorebreaksecurity.com/blog/ssrfs-up-real-world-server-side-request-forgery-ssrf/.
[26] A. Ng. 2021. Amazon tells senators it isn't to blame for Capital One breach. Cnet. Retrieved from https://www.cnet.com/news/amazon-tells-senators-it-isnt-to-blame-for-capital-one-breach/.
[27] T. Spring. 2021. Misconfigured reverse proxy servers spill credentials. Threatpost. Retrieved from https://threatpost.com/misconfigured-reverse-proxy-servers-spill-credentials/132085/.
[28] A. Tiurin. 2021. A fresh look on reverse proxy related attacks | acunetix. Acunetix. Retrieved from https://www.acunetix.com/blog/articles/a-fresh-look-on-reverse-proxy-related-attacks/.
[29] K. Narayan. 2020. How an attacker could use instance metadata to breach your app in AWS. McAfee Blogs. Retrieved from https://www.mcafee.com/blogs/enterprise/cloud-security/how-an-attacker-could-use-instance-metadata-to-breach-your-app-in-aws/.

[30] T. Pohl and B. Williams. 2021. Hacking the cloud: Exploiting AWS misconfigurations. Retrieved from https://www.youtube.com/watch?v=0PhKK-GHgBI.
[31] Capital One. 2019. Capital One Announces Data Security Incident. Retrieved on 24-July-2022 https://www.capitalone.com/about/newsroom/capital-one-announces-data-security-incident/.
[32] E. M. Hutchins, M. J. Cloppert, and R. M. Amin. 2011. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Leading Issues in Information Warfare & Security Research*, vol. 1. Retrieved on 24-July-2022 https://www.lockheedmartin.com/content/dam/lockheed-martin/.
[33] Cloudflare. 2020. What is a WAF? Web application firewall explained. Cloudflare. Retrieved from https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/.
[34] J. Davis. 2020. Capital One CIO: We're a software company. *Information Week*. Retrieved from https://www.informationweek.com/strategic-cio/executive-insights-and-innovation/capital-one-cio-were-a-software-company/d/d-id/1333457?
[35] PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS). 2018. Requirements and Security Assessment Procedures - Requirement 6.6, Version 3.2.1. Retrieved on 24-July-2022 https://www.pcisecuritystandards.org/document_library/?category=pcidss&document=pci_dss.
[36] S. Peters. 2019. What's in a WAF? Dark Reading, Article, November 20, 2019, [Online]. Retrieved 21-Dec-2020 https://www.darkreading.com/edge/theedge/whats-in-a-waf-/b/d-id/1336402.
[37] G. Otto. 2020. What Capital One's cybersecurity team did (and did not) get right. Cyberscoop. Retrieved from https://www.cyberscoop.com/capital-one-cybersecurity-data-breach-what-went-wrong/.
[38] K. Cohn-Gordon, C. Cremers, and L. Garratt. 2016. On post-compromise security. In *Proceedings of the IEEE Computer Security Foundations Symposium*. 164–178.
[39] G. Mateaki. 2020. PCI 6.6: Why you need a web application firewall and network firewall. Retrieved from https://www.securitymetrics.com/blog/pci-66-why-you-need-web-application-firewall-and-network-firewall.
[40] A. Andriotis and R. L. Ensign. 2020. Capital One cyber staff raised concerns before hack. *The Wall Street Journal*. Retrieved from https://www.wsj.com/articles/capital-one-cyber-staff-raised-concerns-before-hack-11565906781.
[41] J. Murdock. 2021. Amazon refuses blame for Capital One data breach, says its cloud services were "not compromised in any way." Retrieved from https://www.newsweek.com/amazon-capital-one-hack-data-leak-breach-paige-thompson-cybercrime-1451665.
[42] AWS. 2021. Shared responsibility model—Amazon Web Services (AWS). Retrieved from https://aws.amazon.com/compliance/shared-responsibility-model/.
[43] L. O'Donnell. 2021. Is AWS liable in capital one breach?. Threatpost. Retrieved from https://threatpost.com/capital-one-breach-senators-aws-investigation/149567/.
[44] AWS. 2021. Add defense in depth against open firewalls, reverse proxies, and SSRF vulnerabilities with enhancements to the EC2 Instance Metadata Service. AWS Security Blog. Retrieved from https://aws.amazon.com/blogs/security/defense-in-depth-open-firewalls-reverse-proxies-ssrf-vulnerabilities-ec2-instance-metadata-service/.
[45] The Duquesne Group. 2021. Mission AWS 2019: "Enable our Customers to Innovate." Retrieved from https://www.duquesnegroup.com/Mission-AWS-2019-Enable-our-Customers-to-Innovate_a347.html.
[46] S. Harris. 2021. Capital One breach—Ramblings of a Unix geek. Retrieved from https://www.sweharris.org/post/2019-08-21-capital-one/.
[47] N. Eide. 2020. Capital One's public cloud strategy at odds with industry. CIO Dive. Retrieved from https://www.ciodive.com/news/capital-ones-public-cloud-strategy-at-odds-with-industry/547245/.
[48] Capital One. 2021. Capital One annual report 2018. Retrieved from https://ir-capitalone.gcs-web.com/static-files/04c57bd9-b351-418c-9f18-ed91d4bfad23.
[49] Capital One. 2021. Capital One annual report 2019. Retrieved from https://ir-capitalone.gcs-web.com/static-files/2f0f821a-0db0-4eab-9895-63013c4e59c2.
[50] J. Surane and L. Nguyen. 2021. Capital One touted the data cloud's safety. Then a hacker breached it. *Los Angeles Times*. Retrieved from https://www.latimes.com/business/story/2019-07-30/capital-one-cloud-safety-hacker-breach.
[51] Capital One. 2022. Press Release. Capital One. Retrieved from https://www.capitalone.com/about/newsroom/capital-one-announces-data-security-incident/.
[52] ISACA/Protiviti. 2019. Today's Toughest Challenges in IT Audit: Tech Partnerships, Talent, Transformation.Audit Benchmarking Study, [Online]. Retrieved on 24-July-2022 https://www.protiviti.com/US-en/insights/it-audit-benchmarking-survey.
[53] Federal Reserve Education. 2021. Banking supervision. Retrieved from https://www.federalreserveeducation.org/about-the-fed/structure-and-functions/banking-supervision.
[54] Comptroller of the Currency, Federal Reserve System, Federal Deposit Insurance Corporation, and Thrift Supervision Office. 2000. Federal register: Interagency guidelines establishing standards for safeguarding customer information and rescission of year 2000 standards for safety and soundness. 65 FR 39471, 26-Jun-2000. Retrieved

from https://www.federalregister.gov/documents/2000/06/26/00-15798/interagency-guidelines-establishing-standards-for-safeguarding-customer-information-and-rescission.

[55] A. Marotta and S. Madnick. 2020. Analyzing the Interplay Between Regulatory Compliance and Cybersecurity (Revised). Working Paper CISL# 2020-15, Available at SSRN: https://ssrn.com/abstract=3569902 or http://dx.doi.org/10.2139/ssrn.3569902

[56] C. Feeney. 2017. Testimony of Christopher Feeney before the US Senate Committee on Homeland Security & Govt. Affairs – 'Cybersecurity Regulation Harmonization', Financial Services Roundtable (FSR) - BITS, 2017. Retrieved on 24-July-2022 https://www.hsgac.senate.gov/imo/media/doc/Testimony-Feeney-2017-06-21.pdf

[57] Kaspersky. 2017. Banks spend on IT security is 3x higher than non-financial organizations. Kaspersky. Retrieved from https://www.kaspersky.com/about/press-releases/2017_banks-spends.

[58] N. Nelson. 2008. A STAMP Analysis of the Lex Comair 5191 Accident. Master's Thesis, Lund University, Sweden. [Online]. Retrieved on 24-July-2022 http://sunnyday.mit.edu/papers/nelson-thesis.pdf.

[59] P. S. Nelson. 2008. A STAMP Analysis of the Lex Comair 5191 Accident. Master's Thesis, Lund University, Sweden. [Online]. Retrieved on 24-July-2022 http://sunnyday.mit.edu/papers/nelson-thesis.pdf.

[60] S. Malmquist, N. Leveson, G. Larard, J. Perry, and D. Straker. Increasing Learning from Accidents – A Systems Approach illustrated by the UPS Flight 1354 CFIT Accident. [Online]. Retrieved on 24-July-2022 http://sunnyday.mit.edu/UPS-CAST-Final.pdf.

[61] N. G. Leveson. 2016. CAST Analysis of the Shell Moerdijk Accident. [Online]. Retrieved on 24-July-2022 http://sunnyday.mit.edu/shell-moerdijk-cast.pdf.

[62] N. Leveson, M. Daouk, N. Dulac, and K. Marais. Applying STAMP in Accident Analysis. [Online]. Retrieved on 24-July-2022 https://shemesh.larc.nasa.gov/iria03/p13-leveson.pdf.

[63] N. G. Leveson and M. Joel Cutcher-Gershenfeld. 2004. What System Safety Engineering can learn from the Columbia Accident. [Online]. Retrieved on 24-July-2022 http://sunnyday.mit.edu/papers/issc04-final.pdf.

Received 11 May 2021; revised 20 March 2022; accepted 22 June 2022