Environments

A How to CISO Handbook

One challenge of being a CISO is understanding *scope*: when a colleague tells you a truth (hypothetically "We patch our systems regularly"), in what environment is that true? Maybe they're just referring to the core Windows Domain servers, or possibly to desktops, but it's easy for executives, including CISOs, to hear that as "we do this everywhere." So what are our environments, and how did we get here?

The Paper-Native Era

Companies have always had multiple environments: the *corporate* environment in which they operate the corporate entity (making sure the lights stay on, employees get paid, etc), and the *production* environment, where they generate revenue. The operators of the *corporate* environment are typically the G&A functions (HR, Finance, etc), and the "customers" are the employees. In the *production* environment, employees operate the environment, and the "customers" are actual customers. In the earliest days, pre-computing, business activities in both of these environments were paper- and telephone-based: accountants did double-entry bookkeeping by hand, credit cards were processed in batches at the end of the day, even as cash was deposited at a bank branch. Security was a *physical* environment problem, keeping adversaries away from sensitive assets, while ensuring integrity against adversarial insiders. Authentication was sometimes a matter of knowing the handwriting of your employees, so you could track down who made which mistake.

The Shift to Computers

In the early days of the IT revolution, enterprises using workstations were few and far between, and quite often those systems were merely terminals that connected to a mainframe. Users would authenticate themselves using a password at a fixed location, and the mental model of the *keyboard* being the edge of the network perimeter became established in the minds of IT architects and the rare security engineer. These systems were usually either strictly corporate, with a set of accounting applications, and possibly a word processor or spreadsheet application, or strictly production, managing inventory and associated work orders, and limited point of sale applications. This gave rise to the first *datacenter* environments, even if in most companies these started as repurposed storage closets.

But even as companies add in compute-based environments, paper- and location-based environments persist. Processes advance by word of mouth; sticky notes keep track of work-in-progress, and sometimes, the best way to understand what's going on in an environment is to walk around and ask people.

Compute-Native To Network-Native

The 1980s saw the adoption of the personal computer, which was rarely networked in any meaningful fashion. Workstations might still have multiple humans using them, who might either share a role-based account, or have individual accounts on a system. Administrators would have separate accounts to maintain multiple systems, often physically going from system to system to update them. Users might share *data*, but applications were more often tied to specific systems. Security tools are limited in this *desktop* environment, with anti-virus leading the way as a sort of "data firewall" when examining data moving on disks.

The 1990s began the adoption of the *networked* enterprise. Administrators could now remotely access and administer systems, and applications for users could be centralized, enabling true multi-user interactions into core IT applications. Application access was no longer one-to-one, but most applications did not yet truly support different roles for different users. Most enterprise networks were isolated from outside interactions, and gaining access to an application simply required being inside the corporate network and knowing where the application lived. Early virtual private networks (VPNs) used telephone modems to allow remote users to literally *dial in* to an enterprise and allow employees on laptops to work from almost anywhere. Corporate and production applications might live in different data centers, but were both accessed from the same user desktop. The network and data-center focus lead to security tools that focus on traffic analysis and perimeter control, with the beginning of centralized administrative tools to manage user desktop security.

The reference architecture enterprises inherited in the 2000s was often described as "crunchy on the outside, and soft and gooey on the inside." Now that enterprises could connect to the Internet, users could access the World Wide Web and begin to use applications in the very beginnings of what would come to be known as the Software as a Service model. At the same time, VPNs moved from telephone connections to remote access over the Internet, and the idea that a corporate network was some form of isolated enclave from the Internet became laughable. System compromises were routine. Malware propagated almost at will, and once an adversary was inside a corporate network, *lateral movement* allowed adversaries to freely explore an enterprise's most valuable information assets. IT and security professionals referred to *infections* and *outbreaks*, rather than *breaches* and *compromises*, and organizations charged into the future with abandon.

Internet of Things devices gained significant traction in corporate buildings – while IT teams were worrying about maintaining infrastructure like printers and routers, facilities teams were deploying badge access systems, rooms wizards, and security cameras. Virtualized networks were used more often than not to attempt to isolate the HVAC controllers from credit card point of sale systems, with … mixed results. In the same way that modern cars can be conceived of as a network of computers that can drive itself, the modern office building became a network of computers that housed itself.

The Rise of Cloud-Native

The 2010s begin with the evolution of the enterprise *away* from the centralized data center, and towards cloud computing. While some *corporate* applications migrate to cloud service providers, the predominant use case was for *production* applications to do so. The ease of rapidly provisioning and deploying applications in a cloud environment revealed one of the greatest weaknesses of modern IT support and cybersecurity teams: they had evolved to provide centralized capabilities to the *datacenter* or *network*, and relied heavily on the latency involved in slow deployments to react to new and novel applications. Once developers could quickly deploy applications *without* some support gatekeeper, they did so at ever increasing paces, relying on the not-so-effective *shared responsibility* model of the cloud providers to ensure secure operations.

Developers quickly discovered that while *cloud* was fast to deploy in trying out new applications and ideas, it was even speedier to deploy production applications. Unfortunately, while cloud providers have an abundance of security infrastructure available to help secure applications in these environments, those tools are rarely well-implemented (either by the cloud providers or the consumer). *Shared responsibility* becomes a hot topic, as cloud providers might provide the *ability* to become secure, but rely on the customers to know enough to protect themselves.

Multi-cloud starts out as a core concept – that customers could use multiple distinct providers to deliver their applications, eliminating vendor lock-in. The reality is far different. Each application takes advantage of the slightly different capabilities of each cloud provider, resulting in applications that don't easily move between two different clouds. Instead, most enterprises' multi-cloud strategy is to deploy different application environments into different clouds, often segregated along *organizational* boundaries, minimizing how many platforms individual development teams have to understand … but increasing the complexity of knowledge required by teams that support across organizational boundaries, including security and operations teams.

Fewer and fewer enterprises are building and deploying applications in their own data centers, both to manage costs (treating computing as opex, not capex), speed up time to market, and create flexibility down the road.

Businesses become SaaSy

Even as datacenter operations moved from bare metal to public clouds, software in support of business operations shifted from running on an enterprise's systems to being consumed as a service. While CRM systems like Salesforce may have been the flagship product leading the way into a Software as a Service (SaaS) world, the ability for an enterprise user to quickly provision an application with just a credit card has more fundamentally transformed business operations. While mega-SaaS platforms have replaced core business operations from HR management to accounting, micro-SaaS platforms have provided capabilities that used to be entirely unsupported: social media management, design, editing, scheduling: almost every employee in your company now has entire SaaS industries aimed at making their job simpler.

IT teams which used to spend the majority of their efforts on supporting the software that underlies an application, with a small focus on application support for major applications, have had to shift to focus on inventory management of third party applications, with a side focus on securely configuring those applications within their enterprise ecosystem.

The Rise of the Corporate Cyborg

While enterprise applications were making their transitions, users were evolving in their own directions. For most organizations, the Blackberry was their first taste of consumer IT coming into the network. Users–first sales reps, but then others–would buy their own devices, and insist on using those instead of approved devices. IT teams, trapped between a cost-cutting mentality from the CFO and early adopters in the business, often shrugged, called it *Shadow IT*, and moved on, providing only the minimum support they could get away with. Over time, more and more users adopted various bring your own device (BYOD) technologies, from smartphones and laptops, to more esoteric (yet still networked) devices. Applications on these devices are continuously communicating on behalf of their user.

IT teams which have focused on "authenticate the user" often overfixate on verifying the *human*, and miss the opportunity to move to authenticating the *cyborg*, and letting the individual devices authenticate the human to their satisfaction.

Tracking Your Environments

Coming back to where we started: It can be useful to keep a quick master list of the interesting environments you're responsible for, as a quick reference when you hear a "truth" about security. Ask "In which environments is that true? Which environments need that to be true? How hard is it to learn whether or not it's true?" Your tracking sheet might contain a list like this:

Enterprise	**Production**	**Users**
Enterprise Cloud Servers	Production Cloud	Corporate Endpoints
Enterprise Data Centers	Production Data Center	User Endpoints
Enterprise Infrastructure	Production Labs/Dev	User Smartphones
Enterprise IOT	Production Build Infrastructure	User IOT
Enterprise SaaS	Production SaaS	User SaaS

For each item, do you have access to an inventory? How hard is it to get the inventory? Those two measures alone will give you insight into how effectively you understand your infrastructure.