SOUNDING THE ALARM: THE TOP MAINFRAME SECURITY THREATS

The headlines keep on coming. British Airways data breach: thousands of customers given more time to claim compensation. Amazon data breach fears, European e-ticketing platform Ticketcounter extorted in data breach, Oxfam Australia supporters embroiled in new data breach. And on and on it goes.

While these breaches don't specifically relate to mainframes, it's clear we are living in dangerous times. Especially when the mainframe continues to be the processing and transactional heart of so many organizations and is increasingly viewed as a hub for innovation. In every location where our team carries out a security audit or penetration test, we always expose significant, previously unknown security concerns. So, why is the mainframe at risk to this degree?

The people who truly "know the mainframe" and understand it have traditionally been a small group. This complexity has led most everyone else to conclude that it's virtually secure by default. It's not. There's also a general lack of understanding around the detail of mainframe security by the very people tasked with keeping it secure. This leads to vulnerabilities.

As an example, individuals may not properly understand the risks involved in giving someone a **superuser** privilege. We have workplaces where employees are given **read** access to everything. But in the mainframe world, if you can read something—especially data—you can copy it. If you can copy, you can download. And if you can download, you can potentially exfiltrate the data.

In addition, the average person can't just buy a mainframe, install the software, and start testing it. The technology is still too costly and too tightly controlled to be reverse engineered. But that's changing as more information is shared online. Knowledge about the platform and how it can be

hacked is becoming more widespread.

With that backdrop in mind, I want to share the top threats we've identified in the course of our work.

1. **Too many users with escalated privileges**. In this situation, the Superuser privilege can be inappropriately used, granting users excessive access to system services and OMVS/USS (Unix System Services) resources and data. This means data can be easily copied, deleted, or held to ransom, and the ramifications can be huge. With **read** access so often the norm, instead, the default access should actually be **none**. This is about applying the principle of least privilege (PoLP).

2. **Privilege escalation vulnerabilities.** Many enterprises grant excessive access to libraries and authorized datasets. This increases the risk of someone accessing your files to elevate their own permissions on the system. That could mean taking yourself from **problem state**, where **normal** user/applications run, to **supervisor state**: a supposedly "protected" and authorized elevated state in which a user has free rein to do all the clever stuff—or to make mischief. In the non-mainframe world, this would be called getting route-level privileges. If bad actors can get to that state, they gain the ability to read and write all data, including memory.

3. **Default passwords and weak password management.** Password insecurity is rife; it's been estimated that it would take a hacker less than a second to crack eight of the ten most commonly used passwords. Password vaults are not commonly used. Organizations should not solely rely on passwords and must ensure strong password controls, avoiding static passwords and ensuring they are changed regularly. For mainframe privileged users, multi-factor authentication (MFA) is a must.

4. **Access to sensitive and cryptographic data.** Additional processes, procedures, and rigor are urgently required around protecting cryptographic data and keys. **Read** access to the database allows it to be copied and downloaded. Data set profiles that are poorly configured allow **read**, **update**, and **control** access. This means data can be copied, updated, or downloaded. Once downloaded, offline password-cracking tools can reveal passwords in the database.

5. **"Faceless" accounts**. This is another possible attack vector for hackers that occurs when the organization needs an account for a system task but there's no real person or actual user associated with it. These accounts often come with system-level privileges. They typically have a password that is rarely changed and, if they do have a password, it's usually easy to guess. Have you protected all of your "faceless" accounts properly and are they appropriately defined?

In most cases, weak controls and inadequate security measures are exacerbated by a number of factors. These can include insufficient headcount; inadequate resourcing or a lack of in-house skills combined with poor system configuration; and processes that are no longer fit-for-purpose or simply not in place. Many sites are running outdated (or have a complete absence of) appropriate security tools and technologies.

The best way to start securing your mainframe is to work towards a *Zero Trust* culture. And when you're tackling a specialist area like this, you may need additional firepower. Few people would attempt to rewire their house or apartment and deal with the dangers of electricity without having the right tools on hand—or, indeed, without being an experienced and accredited electrician. Instead, we pick up the phone or head online to find someone who knows what they're doing.

With so much at stake, it may be time to call in the experts.