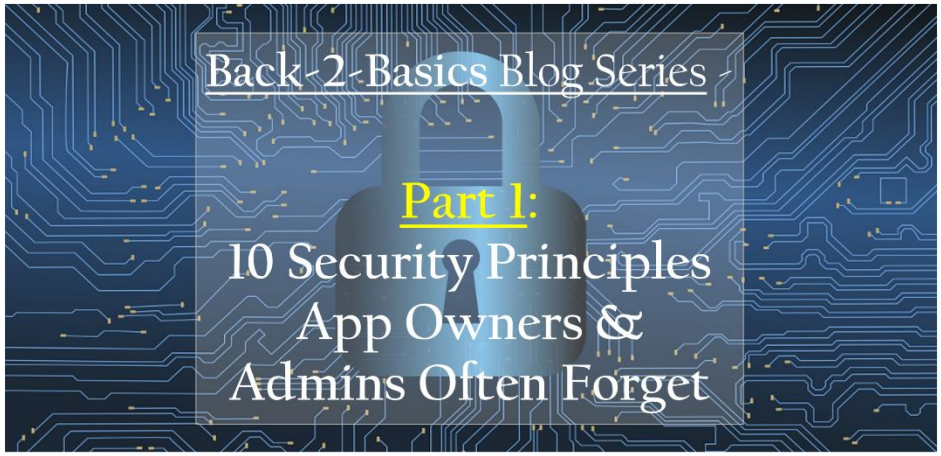
BLOG– Back-2-Basics Blog Series- Part 1: 10 Security Principles App Owners & Admins Often Forget

Posted by: William | On: April 6, 2026 | Back2Basics, Guides & Best Practices, Risk Management
https://wardtechaudit.com/back-2-basics-blog-series-part-1-10-security-principles-app-owners-admins-often-forget/



Back-2-Basics Blog Series – **Part 1:**

10 Security Principles App Owners & Admins Often Forget

Summary Table of Contents

Introduction

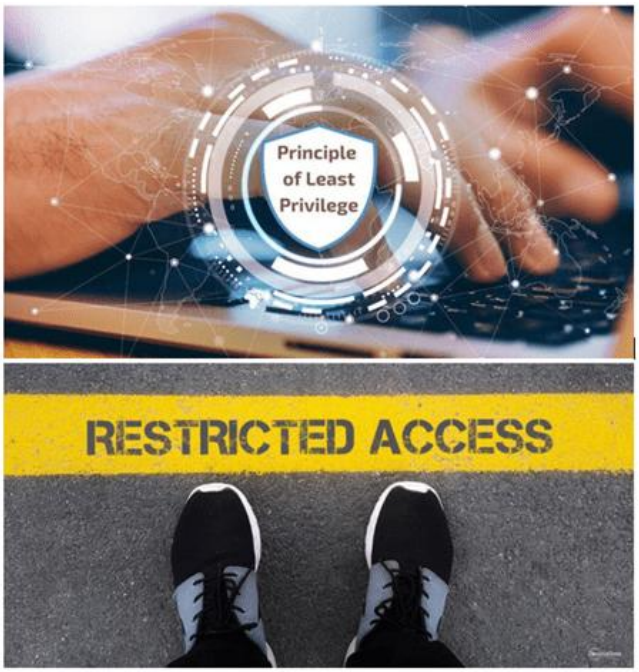
In the rush to adopt the latest tech and flashy tools, many app owners and admins forget the fundamentals of cybersecurity—until a breach reminds them. Before you scramble to patch the newest threats, it pays to step back and master the basics. In this post, we break down **10 foundational security principles** that are often overlooked but make all the difference in keeping your apps, data, and users safe.

1. Principle of Least Privilege

Grant users, processes, and apps only the minimum permissions needed. It minimizes risks by restricting data access to authorized individuals and preventing the spread of malware.

Real-life example: Uber 2022 — a PowerShell script with hardcoded admin creds gave attackers full AWS/Google Workspace access, and other access was identified that was not needed.

Basic mitigation: Hard-coded credentials were removed, exposed admin keys were rotated, multi-factor authentication (MFA) enforced, and stricter secrets management and access controls across AWS and Google Workspace environments were implemented.



2. No Security by Obscurity

Never rely on hidden information, vulnerabilities, URLs, ports, algorithms, etc. rather than using strong, effective built-in security controls. A simple way to explain security by obscurity is: *"It's like hiding your house key under the doormat and hoping no one thinks to look there."* The idea is that something is kept "secure" only because people don't know how it works or where it is—not because it's actually well-protected.

Real-life example: 2016 Mirai botnet hijacked IoT devices with "hidden" default credentials for massive DDoS attacks. DVD CSS encryption collapsed once the algorithm leaked.

Basic mitigation: Enforced strong, unique passwords on all IoT devices, disabled default credentials, and ensured devices were updated with security patches to prevent unauthorized access. Assume everything is discoverable; use strong, publicly vetted mechanisms and run regular external scans/pen tests.

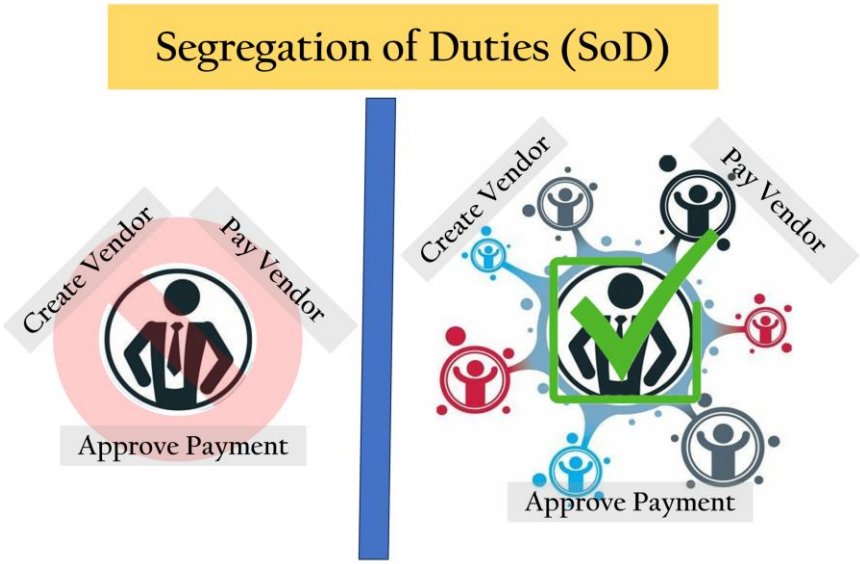
Learn more: What's Lurking In Your Network Folders? Security by Obscurity in practice.

3. Segregation of Duties (SoD)

Divide critical tasks so no single person can complete a high-risk action alone.

Real-life example: Separation of Duties failure occurred at WorldCom in the early 2000s. Certain employees had combined access to create vendors and approve payments. They exploited this to set up fake vendor accounts and approve fraudulent payments. The result was one of the largest accounting scandals in history, with billions of dollars in misreported expenses.

Basic mitigation: Implementation of strict Separation of Duties controls, so that no single employee could both create vendors and approve payments, along with enhanced internal audits, financial oversight, and compliance procedures to prevent similar fraud.

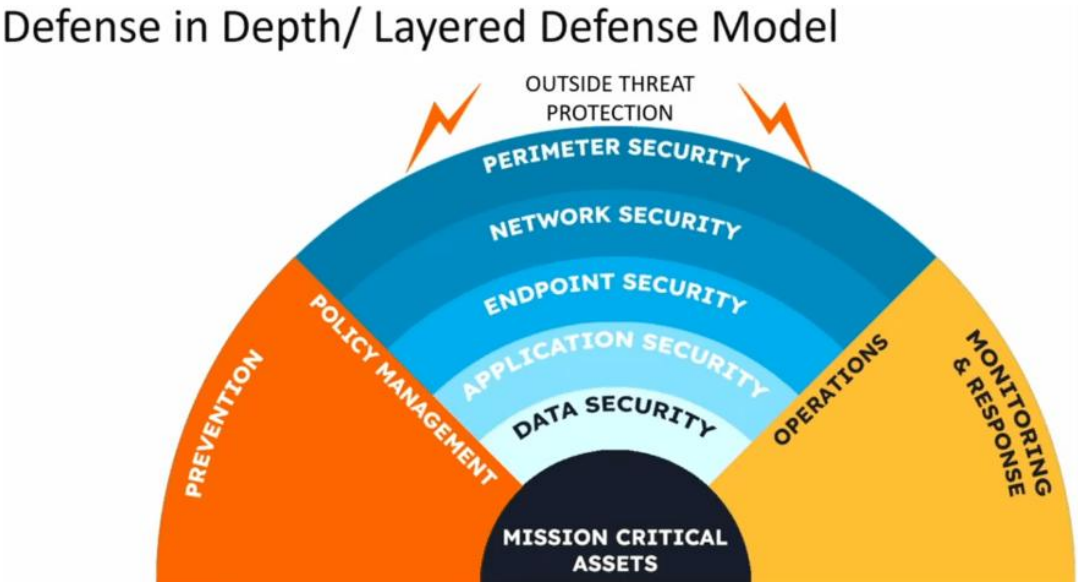


4. Defense in Depth (Layered Security)

Use multiple overlapping controls instead of one single barrier.

Real-life example: Target hack in 2013 — HVAC vendor breach led to payment systems because there was no internal segmentation. Many ransomware attacks succeed with only a firewall or endpoint tool.

Basic mitigation: Layered network segmentation + MFA + detection + immutable backups; tested with red-team exercises.



5. Zero Trust Mindset

Always design as if attackers are already inside—*Verify all access*.

Real-life example: MS Midnight Blizzard 2024 -Initial entry led to persistence because internal traffic wasn't fully verified.

Basic mitigation: Enforced micro-segmentation, continuous authentication, and anomaly detection.

Bonus: Getting Started with Zero Trust: Implementation Steps*

Rolling out Zero Trust security seems daunting, yet you can tackle it step by step. The sequence below gives SMBs a practical roadmap fitting limited budgets and staff.

1. **Map What You Own**: List every user, device, workload, and piece of data that touches your network infrastructure. Record where those network resources sit: in the office, at home, or in cloud environments, and note the data points they process. This baseline inventory helps you spot hidden systems and outdated endpoints before attackers do.
2. **Examine the Current Security Posture:** Run vulnerability scans, privilege audits, and configuration reviews. Compare findings with NIST guidance and pull fresh threat intelligence from CISA feeds. Look for over-privileged accounts, weak passwords, or third-party integrations that increase exposure to supply chain attacks.
3. **Identify Critical Assets:** Rank applications and data by business impact. Ask, "*If this goes offline, how much revenue do we lose?*" That ranking drives the order in which you apply Zero Trust principles and avoids wasting time on low-value targets first.
4. **Draft Context-Based Access Policies:** Create rules that apply least privilege access and the assume-breach mentality. Define conditions such as physical location, device health, and role that must be true before users gain access. Use clear language so admins can adjust policies without breaking workflows.
5. **Strengthen Identity Verification:** Integrate identity providers offering multi-factor authentication and continuous authentication. Each login should continuously verify user identity and device context. This blocks compromised credentials from moving laterally inside the corporate network.
6. **Segment Your Network:** Divide workloads into small, isolated network segments. Micro-segmentation keeps an intruder from hopping between databases and app servers. In cloud services, apply the same logic with security groups or service mesh policies.
7. **Enable Monitoring and Analytics:** Deploy tools that watch network traffic and user behavior 24/7. Real-time alerts let security teams shut down suspicious sessions before attackers reach sensitive data.
8. **Build a Phased Rollout:** Start with one high-value app as a pilot. Measure impact, tune access control, train users, and then extend controls across additional systems. This staged approach fits lean teams and aligns with the trust maturity model promoted in the federal zero-trust strategy.

Follow these eight steps to transform a traditional perimeter into a resilient Zero Trust environment without derailing daily operations. ***Credit to ssldragon.com for *"Getting Started with Zero Trust: Implementation Steps"* (Source: https://www.ssldragon.com/blog/what-is-zero-trust/)**

6. Keep It Simple, Stupid (KISS for Security)

Favor straightforward designs over complex ones.

Real-life example: 2020 SolarWinds supply chain attack.

What happened:

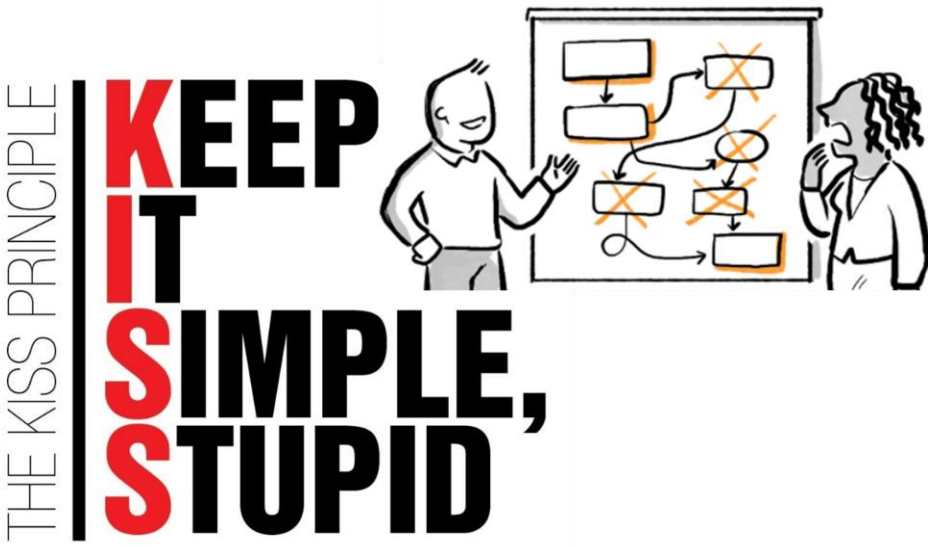
- Attackers compromised the build system and inserted malicious code into updates.
- A very complex software supply chain and trust model was involved.

Where complexity hurt:

- Organizations trusted a deeply embedded tool without verifying updates.
- Hard to validate every layer of a complex supply chain.

Simple takeaway: The more layers of trust you add, the harder it is to verify them.

Basic mitigation: It was mitigated by isolating or removing affected (Orion) systems, applying patched updates, and strengthening software supply-chain security with stricter code integrity checks and monitoring.



7. Default Deny (Fail Securely / Secure by Default)

When something fails, default to the most secure state.

Real-life example: 2019 Capital One data breach – A misconfigured web application firewall allowed unintended access to internal resources. IAM roles then allowed access to sensitive S3 data. Instead of *deny by default*, permissions and network rules were too permissive.

Basic mitigation: They enforced a strict default-deny model—blocking all access by default and only allowing explicitly defined users, systems, and communications that are absolutely necessary. Remember, Deny by default, suppress verbose errors in production, and log internally only.

8. Minimize Attack Surface

Reduce exposed features, ports, APIs, code paths, and unnecessary services—every addition increases risk.

Real-life example: 2024 Snowflake data breach, where attackers gained access to **over 100 customer environments** hosted on Snowflake primarily due to failure of Snowflake to minimize the attack surface. Credentials were targeted that had no Multi-Factor Authentication (MFA). In other words, attackers didn't need a sophisticated exploit—they just found **exposed, over-permissioned access points** and 'walked right in' the front door.

Basic mitigation: They consistently required and enforced MFA, restricted access to only necessary users and data, and eliminated unused or exposed accounts to reduce the overall attack surface.

9. Continuous (Secure) Logging, Monitoring & Audit Trails

Continuously log what's happening in your systems, watching for suspicious activity, and keeping reliable records so you can investigate issues later.

In other words:
Logging: Keep a record of actions (who did what, when)
Monitoring: Actively watch those records for unusual behavior
Audit trails: Maintain trustworthy history to trace and investigate incidents
Tip: Always remember to follow your organization's Data Retention Standards.

Real-life example: 2014 Yahoo data breach where attackers compromised virtually all 3 billion Yahoo accounts.

Basic mitigation: Yahoo enhanced logging, implemented continuous monitoring, enforced real-time alerting on suspicious activity, and now maintain robust audit trails so that anomalous access or data exfiltration is detected immediately. Note best practices also include centralizing logging and (if possible) keeping at least two years of log data.

10. CIA Triad – Protect Confidentiality, Integrity, and Availability

Balance data privacy, accuracy, and accessibility.

The CIA Triad is a core model in cybersecurity that stands for:

– **Confidentiality** – Keep data private and accessible only to authorized users.
– **Integrity** – Ensure data is accurate and hasn't been tampered with.
– **Availability** – Make sure data and systems are accessible when needed.

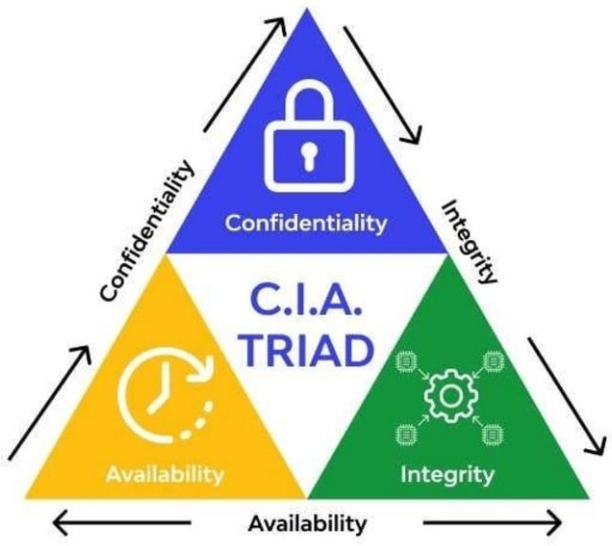
Key benefits:

– Protects sensitive information from unauthorized access
– Maintains trust in data accuracy
– Ensures systems and data are reliable and usable when needed

In short: ***"Keep it secret. Keep it correct. Keep it available."***

Real-life example: Equifax 2017 prioritized availability over confidentiality (147M records exposed).

Basic mitigation: Classified data by CIA needs and include all three in every risk assessment.



To sum up —

Cybersecurity is strongest when organizations follow core, time-tested principles. **Least Privilege** ensures users and systems have only the access they truly need, while **No Security by Obscurity** reminds us that secrecy alone isn't protection. **Separation of Duties (SoD)** prevents a single person from controlling critical processes, and **Defense in Depth** adds multiple layers to stop attacks. A **Zero Trust Mindset** means never automatically trusting anything, and **Keep It Simple (KISS)** highlights that simpler systems are easier to secure. Systems should **Fail Securely** by default, **Minimize Attack Surface** to reduce exposure, and implement **Continuous Logging, Monitoring & Audit Trails** to detect threats early. Finally, the **CIA Triad**—Confidentiality, Integrity, and Availability—provides the foundational goals for protecting data and systems.

Together, these principles form a blueprint for resilient, practical, and proactive security.

---

10 Security Principles – Reference Summary:

1. **Principle of Least Privilege** – Give only the minimum access needed to do the job.
2. **No Security by Obscurity** – Don't rely on hiding things; use real security controls.
3. **Separation of Duties (SoD)** – Split critical tasks so no one person has full control.
4. **Defense in Depth (Layered Security)** – Use multiple layers of protection in case one fails.
5. **Zero Trust Mindset** – Never automatically trust—always verify.
6. **Keep It Simple (KISS for Security)** – Simpler systems are easier to secure and manage.
7. **Fail Securely (Default Deny / Secure by Default)** – Block everything unless it's explicitly allowed.
8. **Minimize Attack Surface** – Reduce the number of ways attackers can get in.
9. **Continuous Logging, Monitoring & Audit Trails** – Always record and watch activity so issues are quickly detected and traceable.
10. **CIA Triad** – Protect data by keeping it private, accurate, and available when needed.