Playbook

Powered by

TuxCare

Top Ten Cybersecurity Misconfigurations

Deploying new cybersecurity tools and processes is just the first step.
Ensuring they are used adequately is the necessary follow-up.

Contents

Abstract

The increasing complexity of cyber threats necessitates robust cybersecurity practices. Among these, the configuration of software and network systems plays a pivotal role in securing or compromising an organization's digital assets. This document draws upon insights from the National Security Agency (NSA) and Cybersecurity and Infrastructure Security Agency (CISA) to highlight the "Top 10 Cybersecurity Misconfigurations" encountered in government and private sector organizations. These misconfigurations, identified through comprehensive red and blue team operations, represent common vulnerabilities that adversaries exploit. Addressing these misconfigurations through strategic mitigation efforts is crucial for enhancing cybersecurity resilience. This document provides an in-depth analysis of each misconfiguration and its real-world impacts while offering pragmatic solutions to bolster cybersecurity defenses.

Target Audience

This book is aimed at IT professionals in Operations, Cybersecurity or Management roles. It provides insights into poorly implemented practices and how to strengthen cybersecurity through the appropriate use of available resources.

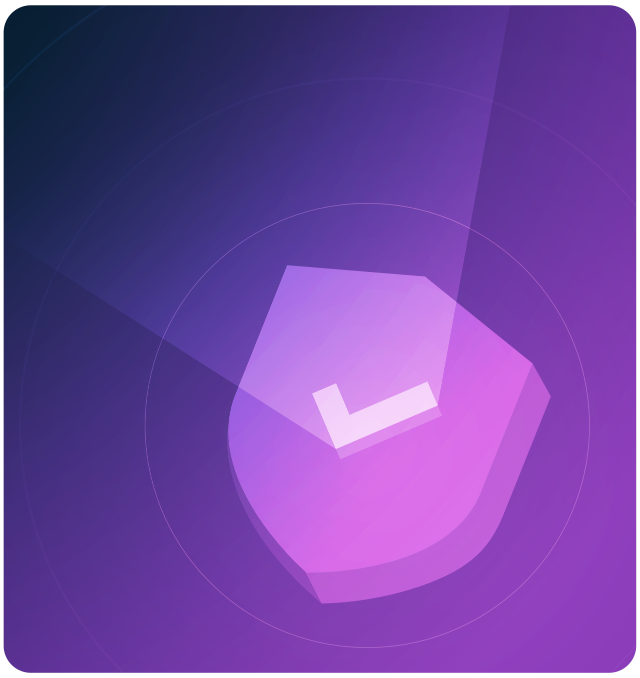
If your organization has deployed any cybersecurity measures, this book offers insights that enable you to assess if those tools are being used correctly or not. Sometimes, the apparent illusion of security by simply ticking a compliance checkbox does not translate to an effective use of more secure practices, and it is not always obvious how that happens.

Go beyond simply deploying MFA to actually ensuring your users (and IT team!) use it everywhere correctly, for example.

Introduction

The relentless evolution of cyber threats, coupled with the increasing reliance on digital infrastructure, underscores the imperative for robust cybersecurity measures. Among the myriad components that constitute a cybersecurity framework, the configuration of software and network systems stands out as both a linchpin and Achilles' heel. Misconfigurations in these domains can open the floodgates to cyber adversaries, leading to data breaches, operational disruptions, and compromised national security.

The importance of addressing cybersecurity misconfigurations cannot be overstated. Recognizing this, the National Security Agency (NSA) and Cybersecurity and Infrastructure Security Agency (CISA) have undertaken red and blue team operations across various government and private sector entities. These operations, aimed at testing and enhancing cybersecurity defenses, have unearthed a consistent pattern of vulnerabilities stemming from common misconfigurations.



This document endeavors to shed light on the "Top 10 Cybersecurity Misconfigurations" as identified by the NSA and CISA. These misconfigurations, ranging from the use of default configurations to inadequate internal network monitoring, not only present a snapshot of prevalent cybersecurity challenges but also serve as a guide for organizations seeking to fortify their defenses. By digging into the specifics of each misconfiguration, this book aims to provide a comprehensive understanding of the threats they pose, the real-world impacts observed, and the mitigation strategies that can be employed to counteract them.

The pursuit of cybersecurity is a dynamic and ongoing process. As such, the insights presented in this document are intended to aid organizations in their quest to navigate the cybersecurity landscape. Through a detailed examination of common vulnerabilities and the presentation of targeted solutions, this book seeks to contribute to the collective effort of enhancing global cybersecurity resilience.

01

Default Configurations of Software and Applications

Often, the default configurations provided by manufacturers are not optimized for security, leaving systems vulnerable to attacks. Default configurations of system services and applications can inadvertently open doors to unauthorized access and malicious activities or facilitate adversarial reconnaissance operations by relying on well-known operational conditions.

Two significant issues are prevalent: default credentials and permissive default service permissions and configuration settings. These configurations, intended for ease of setup and use, often become the weakest link in cybersecurity.

Real-World Impact

When deploying a well-known software stack (for example, LAMP – Linux, Apache, MySQL, and PHP), the resulting system state can easily be inferred and replicated. This makes it easy to recreate a potential target's system and probe for vulnerabilities, instead of triggering monitoring and alert systems on probe attempts by threat actors against the actual systems. It will also make it easier for an adversary to identify potential attack vectors by simply identifying one component and then assuming (correctly) that the entire stack is present.

Default Configuration Issues

In their red/blue team tests, the NSA and CISA found the following common examples:

Default Credentials

Commercial Off-The-Shelf (COTS) network devices often come with predefined default credentials for administrative accounts. These are easily discoverable and exploitable by malicious actors for unauthorized device access, resetting administrative accounts, or leveraging VPN credentials. Furthermore, devices like printers, scanners, and IoT devices, with their loaded privileged domain accounts, can provide attackers with lateral movement capabilities within a network.

In the past, very high-profile incidents occurred, or were exacerbated, by exploiting such default configurations. Botnets targeting IoT devices, in particular, thrive and grow in environments where movement and discovery of new bots is made by exploiting known default configuration states, thus enabling fast and simple spread of the botnet across multiple devices, networks and organizations. When all the instances of a certain network equipment are pre-configured with a built-in set of accepted credentials, it just makes it easier for attackers to move inside the infrastructure.

Insecure Active Directory Certificate Services (ADCS)

ADCS, a critical component in managing Public Key Infrastructure (PKI) within Active Directory environments, can be compromised due to template misconfigurations. For instance, enabling web enrollment without adequate protections can allow attackers to obtain fraudulent certificates, impersonate legitimate entities, and gain unauthorized access to critical systems and data.

Insecure Legacy Protocols and Services

Vulnerable network services such as LLMNR and NetBIOS Name Service in Windows can be exploited through spoofing, poisoning, and relay techniques. These protocols, if left enabled, allow attackers to intercept domain hashes and gain system access, posing a significant threat to Windows environments.

Insecure Server Message Block (SMB) Service

The SMB service, primarily used for file sharing in Windows, is another victim of insecure default settings. The lack of mandatory SMB signing allows attackers to perform machine-in-the-middle attacks, accessing remote systems without needing to capture and crack hashes.

While these results seem to indicate that the problem is more prevalent on Windows-based networks, that is not necessarily the case. The results can simply have been skewed by the available test environments.

Known default configurations can, and in fact do, affect any type of operating system and environment. For example, most Linux distributions will ship with an enabled sshd (secure shell daemon) configuration that does not enforce key-based login (or even enables it) by default, even if that form of authentication is more resilient than traditional login and password. Another common default configuration is the lack of any brute-force protection on the same sshd service, making any newly deployed (and Internet-facing) Linux system immediately a target for armies of bots probing for vulnerable accounts through trial and error.

Mitigation Strategies

To combat these vulnerabilities, it is crucial to modify the default configurations of applications and appliances before their deployment. This includes changing or disabling vendor-supplied default usernames and passwords and enforcing the use of strong passwords.

For ADCS, ensure secure configurations by updating and patching infrastructure, employing robust monitoring and auditing mechanisms, and implementing stringent access controls. Reviewing and restricting permissions on ADCS templates is also essential to prevent unauthorized certificate issuance.

As a more invasive strategy, systems should not be left in a usable state immediately after deployment. When a system is deployed and is immediately usable, then the benefits of changing the configuration will become diminished. Thus, making it so that the default state is not immediately usable would make the need for configuration changing mandatory rather than optional.

While this runs counter to expectations and has some detractors, it would also force IT teams to intentionally apply environment-specific configuration modifications that would make one deployment different from the next, greatly reducing this type of risk. This type of approach, while more effective, thus has some disadvantages – namely the increase in workload and time-to-deploy of new systems.

In short, some possible strategies to mitigate the problem include:

Insecure Server Message Block (SMB) Service

- Change or disable default usernames and passwords before deploying systems. Employ strong passwords and adhere to vendor-provided security guidelines.

Employ Configuration Management Tools

- Use automated tools for configuration management to ensure secure deployment from the outset.

Incorporate Regular Audits and Updates

- Conduct regular audits of configurations and apply necessary updates to address vulnerabilities.

Educate and Train Employees

- Implement comprehensive training programs to increase awareness of the risks associated with default configurations. This is especially important in BYOD environments, where the risk of default configurations comes from foreign devices.

Implement a Non-Usable Default State Policy

- Deploy systems in a non-usable default state to compel IT teams to apply necessary security configurations. Discuss the trade-offs of this approach, such as increased deployment time and potential resistance.

Continuous Monitoring and Logging

- Establish continuous monitoring and logging practices to detect unauthorized changes to configurations. Consider creating alerts for default configurations being present in a system.

02

Improper Separation of User/Administrator Privilege

The separation of user and administrator privileges is a cornerstone of robust cybersecurity practices. However, improper management of these privileges can lead to severe security vulnerabilities. This chapter explores the nuances of this issue and outlines effective strategies to mitigate associated risks.

Issues in Privilege Separation

Excessive Account Privileges

Overly permissive privileges can allow users to access sensitive data or perform actions beyond their role's scope, significantly increasing the organization's risk exposure and attack surface. Examples include employees having access to financial records or system settings irrelevant to their job functions.

Elevated Service Account Permissions

Service accounts, often necessary for applications and processes, are granted elevated privileges. These accounts become prime targets for attackers due to their broad access within a domain. For instance, a compromised service account in a database can lead to data breaches or unauthorized data manipulation.

Non-Essential Use of Elevated Accounts

Using administrative accounts for routine, non-administrative tasks, such as email access or web browsing, unnecessarily expands the attack surface. It creates opportunities for attackers to exploit these accounts for network infiltration and rapid compromise.

Accounts Maintaining Privileges after User Departure

Poor account management often results in accounts retaining privileges after they are no longer needed, such as when an employee leaves the company or changes roles. This oversight can leave dormant accounts open to exploitation.

Real-World Impact

A common consequence of improper privilege separation is system compromise, especially in web hosting environments. For example, if a web server process running Apache/PHP has excessive privileges, any remote exploit providing unwarranted access would inherit these privileges – potentially compromising the entire system.

Historical incidents in web hosting platforms demonstrate the severe consequences of such oversights. By lowering the privileges assigned to the web server process, the risk vector would also be reduced.

Mitigation Strategies

Limit Administrative Privileges

- Minimize the number of administrative roles there are and perform regular reviews of user roles and policies. This practice ensures that only necessary personnel have elevated access, reducing the potential for internal or external exploitation.
- When reviewing roles for privilege maintenance (or not), take the default stance of not maintaining them, and then validating why they should have those privileges. If there is no valid reason, then the role should have the related privileges removed.

Implement Time-Based Access

- Utilize just-in-time access methods to provide elevated privileges only when necessary, thereby adhering to the principle of least privilege. This approach reduces the time window in which elevated privileges are available, diminishing the risk of their misuse.

Enforce Least Privilege Principle

- Regularly audit user accounts to ensure that they possess only the necessary privileges for their roles. This practice minimizes the potential for unauthorized access or actions within the network.

Restrict Non-Essential Use of Elevated Accounts

- Prohibit the use of administrative accounts for general tasks. Implementing user behavior analytics (UBA) can help detect and prevent such misuse by monitoring for abnormal activities.

Enforce Strict Account Lifecycle Management

- Tightly couple account lifecycles with business requirements. Promptly remove or downgrade accounts when an employee leaves or a contractor's engagement ends, minimizing the risk of dormant accounts being exploited.

Implement Comprehensive Activity Logging

- Enforce strict logging of all privileged activities, ideally across all systems. Use log aggregation tools or monitoring solutions for a holistic view of the infrastructure, aiding in the early detection of unauthorized or anomalous activities.

03

Insufficient Internal Network Monitoring

Effective internal network monitoring is crucial for detecting and mitigating cyber threats. Insufficient monitoring can leave organizations vulnerable to undetected adversarial compromises. This chapter explores the implications of inadequate network monitoring and provides strategies for enhancing monitoring capabilities.

The Problem with Insufficient Monitoring

Limitations in Host and Network Sensor Configurations

Many organizations fail to optimally configure host and network sensors for traffic collection and end-host logging. This inadequacy can lead to undetected adversarial compromise and limits the capability to collect traffic data necessary for establishing a security baseline and detecting anomalies.

Challenges with Host-Based Monitoring Only

Relying solely on host-based monitoring can be insufficient, as this approach informs about adverse activities on individual hosts but not about activities traversing between hosts. For example, an organization with host-based monitoring could identify infected hosts but not the source of the infection, hindering efforts to prevent future lateral movements and infections.

Additionally, a lack of proper monitoring correlation between hosts makes it difficult to understand patterns, either pre or post exploitation, which leads to inefficiencies and potential oversights when attempting to rectify the situation.

Failure in Detecting Lateral Movement and Command and Control Activities

Organizations with insufficient network monitoring might fail to detect lateral movement and command and control activities within their networks. Even mature cybersecurity postures can be compromised if network monitoring is not comprehensive, as demonstrated by assessment teams gaining deep access without triggering security responses.

Mitigation Strategies

Establish a Baseline of Applications and Services

- Regularly audit the access and use of applications and services, especially for administrative activities. This practice helps in understanding normal network behavior, thus making it easier to spot anomalies.

Implement Comprehensive Network Monitoring

- Combine host-based monitoring with network monitoring to gain a full view of the cybersecurity landscape. This combination allows for the detection of malicious activities both on individual hosts and as they move across the network.

Regular Audits of Network Traffic

- Conduct routine audits of network traffic and patterns to identify unusual activities or unauthorized access attempts. This proactive approach aids in early detection and response to potential threats.

Use of Advanced Analytical Tools

- Employ advanced analytical tools and algorithms to analyze network traffic for patterns indicative of cyber threats. Machine learning techniques can be particularly effective in detecting sophisticated attacks that might evade traditional monitoring methods.

Train Staff on Network Monitoring Practices

- Ensure that IT staff are well trained in network monitoring practices and aware of the latest cyber threats. Regular training sessions can help in keeping the team updated on new technologies and attack vectors.

Create Incident Response Plans

- Develop and regularly update incident response plans that include procedures for responding to anomalies detected through network monitoring. These plans should involve both technical responses and communication strategies.

Enforce Adequate Segregation of Hosts

- If there is no reason for two hosts to communicate, then moving each to separate network segments should be considered. While not a monitoring policy per se, this at least attempts to reduce visibility and exposure for threat actors.

Change Block Rules to Block-and-Log

- Identify and investigate blocked attempts to communicate between separate systems. Simply blocking the connection can keep probing attempts undetected for long periods of time.

04

Lack of Network Segmentation

Network segmentation is a critical cybersecurity practice that involves creating security boundaries within a network. The lack of proper network segmentation can leave an organization vulnerable to widespread security breaches. Here, we address the risks associated with inadequate network segmentation and outline strategies for effective implementation.

Risks of Inadequate Network Segmentation

No Security Boundaries

Without network segmentation, there are no defined security boundaries between different network areas, such as user, production, and critical system networks. This lack of segregation allows attackers who compromise one part of the network to move laterally across various systems uncontested.

Increased Vulnerability to Ransomware and Post-Exploitation Techniques

Unsegmented networks are significantly more susceptible to ransomware attacks and post-exploitation techniques. The absence of compartmentalization means that once an attacker gains access, they can potentially exploit the entire network.

Risk to Operational Technology (OT) Environments

Lack of segmentation between IT and OT environments places OT at risk. Despite assurances of air-gapped networks, assessment teams have often gained access to OT networks through overlooked or accidental network connections.

Difficulty in Establishing Monitoring Checkpoints

If all systems can communicate with all other systems, it is virtually impossible to properly log and monitor activity between them. Adopting an architecture with clearly defined checkpoints where monitoring probes can be deployed increases the security and makes it possible to spot anomalous activities sooner.

Mitigation Strategies

Implement Next-Generation Firewalls

- Use next-generation firewalls for deep packet filtering, stateful inspection, and application-level packet inspection. This practice helps to deny or drop traffic that does not align with allowed application protocols, thereby limiting an attacker's ability to exploit network vulnerabilities.

Engineer Network Segments

- Isolate critical systems, functions, and resources by engineering network segments. Implement both physical and logical segmentation controls to create distinct security zones within the network. This isolation helps in containing breaches within a specific segment, preventing them from affecting the entire network.

  When starting to implement network segments, an acceptable starting point is segmenting concerns according to business responsibilities – for example, separating different departments' systems from each other. However, systems handling cross-cutting concerns, like identity management, need to operate across these boundaries, or be implemented in such a way that effective delegation of responsibilities is handled by different servers, in different segments. For example, imagine having separate domain controllers on each segment, only talking to one another and to the systems on their respective segments, rather than a single domain controller reachable by all systems on all segments. Properly weighing the pros and cons of extra separation but higher administrative overhead is a must.

Establish Strong Access Control Measures

- Implement strict access control measures for each network segment. Ensure that only authorized users and systems can access sensitive areas of the network.

Regular Audits and Monitoring

- Conduct regular audits and continuous monitoring of network segments to detect any unauthorized access or anomalies. This practice is essential for maintaining the integrity of the segmented network.

Training and Awareness

- Train staff on the importance of network segmentation and its role in cybersecurity. Ensure that they understand the procedures for maintaining and monitoring segmented networks.

Integration with Incident Response Plans

- Ensure that your incident response plans consider the segmented network architecture. This integration helps in quicker containment and response to security incidents within segmented areas.

05

Poor Patch Management

Even years after being widely reported and known, systems vulnerable to the Log4j exploit are still accessible online. This enduring vulnerability highlights a critical failure in patch management practices. Systems that remain unpatched against such well-known vulnerabilities are, in essence, open doors for cybercriminals, often leading to swift and uncompromising exploitation.

The log4j example is just that – a very loud example of poor patch management and its direct impact on the cybersecurity positioning of organizations. The truth is that the lack of a consistent, reproducible, auditable and responsive patch management strategy is still the root cause of very costly incidents and business problems. Maybe, just maybe, the practices used 20 odd years ago are no longer applicable to a threat landscape that has evolved beyond the scope of what was once considered an efficient patch deployment timeframe.

Reasons Behind Poor Patch Management

Lack of Approval from Higher Management

Often, there's a delay or outright refusal to approve downtime for patching due to business continuity concerns. While this is defensible from a business perspective, it fails to account for the incurred losses and disruptions caused by not doing it in the first place, which will (always?) far outweigh the disruption happening during a planned and scheduled maintenance window.

Disruption Concerns

The reboot of systems or services post-patching can lead to significant business disruption. And if disruption is the main reason not to approve timely response to emerging threats, there are also solutions to address it, as listed below.

Prioritization Challenges

Deciding what to patch first within narrow maintenance windows is a daunting task, especially in complex IT environments. The solution to this particular concern is akin to Columbus Egg – don't prioritize at all, and adopt a patching strategy where you simply patch everything. Criticality and severity of different vulnerabilities are often environment specific, or at the very least heavily influenced by environmental variables uncountable during evaluation and scoring. So, even when prioritizing just the highest risk vulnerabilities, you may still be missing others that are relevant to a particular setting.

Resource Limitations

In many cases, the lack of adequate resources hampers the ability to implement timely patches. When working with severely resource-constrained teams, automation is the most efficient way to mitigate the problem. Automate every aspect of the patch management process – testing, deployment, prioritization, and reporting – and you can lower the required resource cost. While not a magical solution per-se, it is a step in the right direction.

The Consequences Are Real

Regardless of the reasons, an unpatched system remains a vulnerable one. The inclusion of poor patch management in the NSA/CISA report underlines its significance. The consequences of inadequate patch management are not hypothetical; they are a real and present danger to organizational security.

Mitigation Strategies

If traditional patch management practices are failing, as evidenced by ongoing vulnerabilities, it's time for organizations to consider alternative approaches:

Live Patching

- A method that allows for disruption-less, efficient, and rapid deployment of patches. This approach can significantly reduce the window of vulnerability without impacting business operations.

Automated Patch Management

- Implementing automated systems for patch deployment can ensure timely updates and reduce the human resource burden.

Regular Security Audits

- Conducting frequent audits to identify and prioritize vulnerabilities can streamline the patch management process.

06

Bypass of System Access Controls

Centralized identity management systems, recommended for reducing authentication risks across multiple platforms, inadvertently create a vulnerability. These systems, which confirm an account's status and access level, exchange messages that are often encrypted but not immune to interception. Weak encryption, shared secrets, or vulnerable third-party systems can allow malicious actors to observe and capture authentication messages. They might then "replay" these messages to gain unauthorized privileges.

There is also the lesser confirmed form of self-inflicted bypass of system access controls, where IT teams have special accounts that are exempt from conforming to stricter requirements enforced on regular accounts. Usually created to facilitate operations, they are a significant risk and, if nothing else, make it obvious that there is a fundamental misunderstanding of the cybersecurity risks in today’s IT environment. These accounts are the goal of any attacker and are usually identified early in any reconnaissance phase prior to an attack. In the situations where they are created to respond to a tool’s shortcomings (for example, a specific management tool has no way of interacting with Multi-Factor Authentication so a special account without MFA is needed) are usually more telling of a larger underlying problem with the tool itself.

How Does This Work Exactly?

Centralized Authentication Woes

It is recommended that organizations run centralized identity management systems to reduce the risk associated with having authentication and authorization spread over multiple systems, any one of which thus becoming a liability in case of an incident. However, this centralization of processes leads to a necessary mechanism through which a third party system asks for confirmation of a given account’s status – authorized or not – and access level (i.e., the privileges that should be awarded after login).

This mechanism relies on message exchange between the centralized identity management system and the third-party system, through a network and usually encrypted. However, due to multiple factors – from weak encryption to shared secrets or improperly protected third-party systems – a malicious actor can observe the authentication process as it happens and collect the resulting message. In some scenarios, it is then possible to “replay” that resulting message and trick systems into accepting it implicitly, without further checks, and grant unwarranted privileges to the malicious actor or to processes under his or her control.

The Active Directory Example

Active Directory's Kerberos authentication is a notable case where hash values are exchanged between systems, leading to vulnerabilities. Here, exploits like "Golden Ticket" and "Pass-the-Hash" have become prevalent in cybercriminal arsenals:

Pass-the-Hash Attack

- This technique involves capturing a hash (a fixed-size alphanumeric representation of a password or key) used in the authentication process. Attackers then use this hash to authenticate themselves and gain access to network resources, bypassing the need for the actual password.

Golden Ticket Attack

- This attack targets Kerberos, a network authentication protocol used in Windows environments. Once attackers gain access to a key domain controller, they can create a 'golden ticket,' which grants authentication as any user within the domain, often leading to unrestricted access across the network.

So I'm Perfectly Safe When Running Linux?

These types of attacks are not specific to any operating system or platform. They also come in many forms, Kerberos just being one example. If nothing else, the fact that centralized identity management supplied by Active Directory is (or can be) consumed in different ways in Linux systems inevitably exposes those systems to the same problem.

Also, the reverse is also possible: you can find Active Directory stand-ins running on Linux providing ldap connectivity and authentication to Windows-based systems. The risk should not be neglected, regardless of whether you're running Active Directory on Windows or not.

In fact, this problem isn't even specific to Active Directory environments either, or even to the internal network and systems. Bypassing System Access Controls can also be achieved, for example, through improperly secured authorization tokens on popular online platforms, such as code hosting, email, and social media services. If a token is created that provides access to a given asset and then that token is stolen, leaked, or accidentally published online, whoever has access to it can obtain the privileges associated with the token without having to re-authenticate with the service, thus bypassing whatever access controls are in place.

Mitigation Strategies

Effective defense against access control bypass requires a holistic strategy:

Enforced Single-Use Privileges

- Authentication should be a one-time process for each session. Once a session ends, re-authentication must be mandatory.

Geo-Location Validation

- Unusual login locations should trigger immediate re-authentication, a crucial step in preventing session hijack attacks.
- A further refinement can even check for the origin client system's location and ip, rather than the specific user. A change in the usual device from where the connection initiates and a re-authentication should be triggered.
- This type of event should be thoroughly monitored for and high risk alerts should be in place when it happens.

Strong Encryption Standards

- Ensure communication between systems using the strongest mutually supported encryption – avoiding default, weaker standards.

Regular System Updates

- Modern systems with updated services can better counteract these attacks, necessitating an ongoing commitment to technological updates.

07

Weak or Misconfigured Multi-factor Authentication (MFA) Methods

The implementation of robust multi-factor authentication (MFA) methods is a pivotal line of defense in cybersecurity. Yet, advisories from the National Security Agency (NSA) and Cybersecurity and Infrastructure Security Agency (CISA) highlight that weak or misconfigured MFA methods remain a prevalent vulnerability. Here, we explore MFA's criticality, common misconfigurations, and best practices to ensure its effectiveness.

Why MFA Is Important

MFA significantly enhances security by requiring multiple verification forms, crucial in an era of sophisticated cyber threats. Password-only authentication has proven inadequate, with services like "Have I Been Pwned" revealing massive data breaches, often including passwords.

MFA introduces an additional layer, combining "something you know" (a password) with "something you have" (like a phone or physical token), fortifying defense against unauthorized access.

Common Pitfalls and Misconfigurations

Over-Reliance on Basic Methods

- SMS or email-based verifications are vulnerable to interception. Exploring more secure alternatives, like app-based tokens or biometrics, is advisable.

Incomplete Coverage

- Neglecting MFA for any account, especially administrative ones, leaves glaring security gaps.

Similar Factor Vulnerabilities

- Using two factors stored in the same location (like a password and a locally stored digital certificate) doesn't constitute true MFA, as both can be compromised simultaneously.

Replicable Factors

- Factors like RFID or NFC tags, susceptible to duplication, offer weaker security. Using them in combination with more secure factors is recommended.

Inadequate User Education

- Educating users on potential attack vectors and their mitigation is vital for maintaining MFA's integrity.

Real-World Implications: The Microsoft "Midnight Blizzard" Incident

A prime example of MFA's importance is the "Midnight Blizzard" attack on Microsoft. Detailed in Microsoft's Security Response Center blog and reported widely, the incident involved unauthorized access to board members' email accounts, facilitated by a lack of MFA.

This breach highlights the need for robust MFA to protect sensitive information and systems. It also exposes the “chink in the armor” problem with cybersecurity, as any gap – however small – can lead to successful breaches and exploits.

Best Practices for Robust MFA Implementation

Diverse Authentication Factors

- Employ a mix of passwords, security tokens, and biometric verifications to fortify defenses.

Comprehensive Coverage

- MFA must be mandatory across all accounts and systems where authentication is required. When starting to deploy MFA, focus particularly on administrative access, but plan to expand coverage rather than stopping there.

Regular Audits and Updates

- Continuously monitor and update MFA configurations to address emerging threats and technological advancements. New methods to attack independent factors emerge regularly, so be prepared to switch away quickly from factors rendered insecure. Avoid considering one factor more important than others.

User Training and Awareness

- Regular education on MFA's importance and secure usage is essential. Emphasize that security should take precedence over convenience and obviate why it’s an existential requirement that it happens.

No Exceptions Policy

- Any account or system without MFA can be a potential entry point for attackers. Ensure universal MFA implementation.

Third-Party Integrations

- Make MFA support a non-negotiable requirement in the procurement process for software and hardware.

Emerging Technologies in MFA

Looking ahead, emerging technologies in MFA, such as advanced biometrics and behavioral analytics, promise to further enhance security measures. Keeping abreast of these developments is crucial in adapting to evolving cyber threats.

08

Insufficient Access Control Lists (ACLs) on Network Shares and Services

Access Control Lists (ACLs) are fundamental in defining access permissions to network shares and services. Yet, as the National Security Agency (NSA) and the Cybersecurity and Infrastructure Security Agency (CISA) report, insufficient or poorly configured ACLs are a widespread cybersecurity misconfiguration. Here, we explore the critical role of ACLs, their common challenges, and strategies to strengthen security through effective ACL management.

Understanding the Role of ACLs in Cybersecurity

ACLs are essential for maintaining data integrity and confidentiality. Insufficiently managed ACLs can lead to unauthorized access and data breaches, making them prime targets for malicious actors.

ACLs can be as granular or high level as necessary – either specifying individual users or entire groups of users. This is useful for fine-grained control, or to ease administration of ACLs at the organizational level. They often reflect, in some fashion, the organizational structure (departments or teams will correspond to specific groups, for example).

The Administrative Aspect of ACLs

Managing ACLs involves a significant administrative effort. For instance, modifying ACLs to reflect changes in access requirements can be a complex process, necessitating either manual updates or automated systems. This overhead can sometimes lead to over-permissive access in the interest of convenience, creating potential security vulnerabilities.

Real-World Risks and Misconfigurations

Unauthorized Data Access

- Attackers exploit ACL weaknesses to access sensitive data on shared drives, using tools like share enumeration or custom malware.

Ransomware Threats

- Ransomware actors employ vulnerability scanning tools to identify and exploit open-access shares.

Convenience Over Security

- Overly broad permissions due to administrative convenience can lead to inadvertent security gaps.

Inadequate Removal of Access

- Often, systems are more efficient at granting than revoking access, leaving potential security risks when users leave an organization.

Best Practices for Effective ACL Management

Least Privilege Model

- Implement ACLs based on the least privilege principle, granting only necessary access.

Regular Audits and Updates

- Continuously review and update ACLs to align with evolving organizational roles and needs. Do look for critical information stored in accessible shares, like credentials or other privileged information.

Comprehensive Monitoring

- Maintain detailed logs for effective detection and investigation of security incidents. Avoid any exceptions (for example, do log administrative changes) to ACLs.

Automated ACL Management

- Utilize software or tools for ACL management to minimize human error and streamline processes.

Centralized ACL Management

- In addition to automation, ACL management should be centralized to facilitate management of permissions and accesses in a single location, rather than spread across multiple systems.

Group-Based Rules

- Manage ACLs based on user groups to simplify administration and ensure consistent access control.

Thorough Documentation

- Keep detailed records of ACL rules, including their purpose, creation date, and author for easier management and review.

Ease of Integration of Existing ACLs with Third-Party Systems

- New systems providing shared resources should integrate with the least amount of friction into the existing infrastructure, rather than requiring customization or special-case scenarios. This requirement will lead to less bugs in the security profile of the organization.

Emphasize a Strong Culture of Security

- More than other misconfigurations, ACLs are vulnerable to convenience shortcuts, as those originate mostly within the IT team. Training and regular reviews should be required.

09

Poor Credential Hygiene

The importance of credential hygiene, encompassing passwords, usernames, and other forms of authentication, helps achieve a robust cybersecurity posture. Despite its role, insufficient credential hygiene remains a significant vulnerability, leading to severe security breaches and data loss. Here, we explore the risks associated with poor credential hygiene and outline best practices for maintaining secure credentials.

Understanding Credential Hygiene and Its Importance

Credential hygiene encompasses practices for securing authentication data, including the creation, storage, and management of passwords. Weak or reused passwords, failing to update credentials, and inadequate system-side security measures like encryption can expose organizations to risks such as unauthorized access, network infiltration, and reputational damage.

The full scope of credential hygiene also covers system-side considerations like encryption levels and storage methods. Relying on cleartext (or reversible) storage of credentials or cleartext transmission of credentials between systems are also tell-tale signs of poor credential hygiene.

Real-World Consequences of Poor Credential Hygiene

Poor credential hygiene has been central to many high-profile breaches. Attackers exploiting weak credentials can escalate privileges or move laterally within networks, leading to data exfiltration and operational disruptions.

On the flipside, the mismanagement of credentials, like requiring overly complex credentials or too frequent changes can also result in user dissatisfaction and insecure practices, such as writing passwords on post-it notes or creating predictable patterns when generating new ones.

Storing passwords in cleartext or in an easily crackable form is extremely risky. The large number of stolen and distributed passwords only makes this problem more visible.

Best Practices for Enhancing Credential Hygiene

Strong and Unique Passphrases

- Passphrases, easier to remember and often more secure, should include a mix of characters, numbers, and symbols.

Avoid Reusing Passwords

- Unique passwords for each account mitigate the risk of multiple account compromises from a single breach.

Use of Password Managers

- These tools alleviate the burden of remembering multiple passwords and enhance security.

Regular Password Changes

- Regular updates are crucial, though excessive changes can be counterproductive. According to Packetlabs, frequent mandatory changes may lead users to create weaker passwords or predictable patterns.

Implement Multi Factor Authentication (MFA)

- MFA significantly enhances security. Google reported that adding a mobile phone for verification blocked up to 100% of automated bots, 66% of targeted attacks, and 99% of bulk phishing attacks.

Educate on Cybersecurity Awareness

- Training in best practices, including the risks of phishing and secure password creation, is essential for improving credential hygiene.

Audit Credential Security

- Disallow storage of cleartext or easily attackable credentials in any system. Keep credential storage systems secure and thoroughly patched.

Maintaining strong credential hygiene is a vital component of an organization's cybersecurity strategy. Adopting practices like using strong, unique passwords, utilizing password managers, and implementing MFA can substantially reduce the risk of cyberattacks and protect sensitive data.

Always consider that adequate credential hygiene can only be achieved when the users understand its importance. Mandatory requirements that exploit complexity to the extreme or impose changes too frequently are just as problematic as the opposite, but will hurt the user's perception more.

10

Unrestricted Code Execution

Unrestricted code execution presents a formidable threat. It occurs when systems allow unverified programs or scripts to execute without adequate restrictions. This issue can be exploited by threat actors to run arbitrary, malicious payloads within a network – often leading to severe security breaches. Here, we explore the nature of this risk and discuss strategies to mitigate it.

Unrestricted Code Execution

Unrestricted code execution allows threat actors to execute arbitrary code after gaining initial access to a system, such as through a successful phishing attack or through exploiting an unpatched vulnerability. Commonly, attackers convince users to execute code that provides them with remote access to internal networks. This code often takes the form of unverified programs or scripts with no legitimate business purpose. These malicious programs often use sophisticated techniques to obfuscate their true nature and bypass security protocols.

Real-World Consequences of Poor Credential Hygiene

- Attackers often use executables, DLLs, HTML applications, and scripts in multiple languages like PHP, ASP, and JavaScript.
- Scripting languages, commonly used in web services – but not restricted to those services – can be manipulated to execute malicious activities without triggering basic security alerts.
- Known vulnerabilities in system drivers can be exploited to execute code at the kernel level, leading to full system compromise.

Mitigating the Risk

Application Whitelisting

- Implement allowlisting to restrict applications and code that can run on the network. Only known and trusted software should be allowed. Note that whitelisting applications is orders of magnitude more secure than blacklisting known-bad executables (a notoriously error-prone and easily manipulated technique).

Regular Security Audits and Monitoring

- Conduct audits to detect and rectify misconfigurations or unauthorized applications. Establish a baseline "expected" profile for systems and monitor for deviations. This proactive approach can quickly identify and address unauthorized applications or processes.

User Education and Phishing Awareness

- Comprehensive staff training can significantly reduce the risk of phishing attacks and other social engineering tactics. Train staff to recognize and report phishing attempts and suspicious activities like unexpected slow performance or application alerts.

Use of Sandboxing Techniques

- Deploy sandboxing to isolate and test untrusted programs and code in a secure environment. Run new software in secure environments, like locked down virtual machines, while assessing their security profile.

Strict Access Control

- Enforce strict access controls to limit what code can be executed, especially by non-administrative users.

Regular Software Updates

- Keep all systems and applications updated to patch known vulnerabilities that could be exploited.

Network Segmentation

- Although primarily a damage mitigation technique, network segmentation can limit the spread and impact of any malicious code that does execute.

Proactive Measures Round-Up

It's easy to notice a pattern in the recommendations for all the misconfigurations analyzed so far: there is a series of recommendations common to most, if not all, of the different misconfigurations. If you're looking for a "shopping list" of practices to adopt, then the following lists the most common suggestions:

Incorporate Regular Audits and Updates

- Identifying weaknesses before they become problems is the best way to proactively increase the security profile of your organization. Red&blue team operations are key in gaining an edge on would-be attackers.

Educate and Train Employees

- The only way to have effective measures in place is to make sure everyone understands why they are necessary and the consequences of not adhering to them. Explaining it, repeatedly and consistently, to everyone in the organization is paramount to ensure they are used effectively and efficiently.

Continuous Monitoring and Logging

- Make sure your infrastructure is set up in a way that enables your team to be the first to be alerted to emerging problems.

Comprehensive Coverage

- The weakest link is always the sought prize for an attacker. If there are any systems or processes with lower security requirements than others, then those are the most likely candidates to be exploited. Don't rely on obscurity for security, as even the most remote and out-of-the-way server/API/network will be found.

Automate everything, everywhere

- Ensuring everything is configured correctly every time requires automation. It's very easy to forget some security features when performing a task manually, so having consistent automation performing these tasks ensures that all the systems have the same security profile, and facilitate the replication of any hardening change across the infrastructure.
- This includes patching automation. Having a patch deployment process that is automatable avoids situations where all the systems are up-to-date except one or two forgotten ones.

Bear in mind that each environment is different. A “best practice” proffered to be the end-all and be-all solution to all your cybersecurity problems might not apply to your exact scenario, or you may have to refine each suggestion to fit your particular set of requirements.

TuxCare

www.TuxCare.com