Title page dal dena report wale type ka

Table of content bhi

**Executive Summary: Cybersecurity Assessment at GitHub**

This executive summary encapsulates the essence of a thorough cybersecurity assessment conducted at GitHub, a leading software organization pivotal to global software development collaboration. The evaluation encompassed 70 information security-related questions, serving as a litmus test for GitHub's adherence to defensive principles and its efforts to mitigate cybersecurity risks. GitHub's commitment to robust defensive measures, platform security, incident response preparedness, and community security awareness emerges as a hallmark, positioning the organization as a frontrunner in the cybersecurity landscape.

In acknowledging the dynamic nature of cybersecurity, this assessment illuminates GitHub's proactive stance in addressing emerging threats. The imperative for continuous monitoring and mitigation efforts is underscored, recognizing that cybersecurity risk mitigation is an ongoing process. GitHub's resilience and commitment to cybersecurity, as revealed in the responses to the 70 questions, lay a strong foundation for sustained vigilance and adaptability in an ever-evolving threat landscape.



**70 Layers of Defense**

There are 70 questions that can be asked to determine whether an enterprise has most defensive principles covered and has taken steps to reduce risk (and entropy) associated with cybersecurity.

1. **Training:**
   1. **Question 1: Do you conduct robust and frequent end-user cybersecurity awareness training?**

GitHub takes cybersecurity awareness seriously, implementing robust and frequent training programs for all employees. These programs cover a range of topics, including the latest cyber threats, social engineering tactics, and best practices for maintaining a secure digital environment. Training materials are regularly updated to reflect emerging threats and industry trends.

GitHub's training approach involves a combination of online modules, workshops, and simulated scenarios to ensure that employees are well-equipped to identify and respond to potential security risks. The training is not a one-time event but rather an ongoing process to reinforce security awareness among the workforce.

**Question 2: Have you taught everyone how to securely store passwords or passphrases?**

GitHub places a strong emphasis on secure password practices. Employees are educated on the importance of using strong, unique passwords and are provided with guidelines for creating and managing secure credentials. The organization encourages the use of password managers to store and generate complex passwords securely.

Multi-factor authentication (MFA) is enforced across all GitHub accounts, adding an extra layer of protection. GitHub regularly communicates updates on password security best practices and ensures that employees are aware of the potential risks associated with weak or compromised passwords.

**Question 3: Do you conduct quarterly anti-phishing, smishing, and vishing campaigns?**

GitHub runs quarterly simulated campaigns to test and enhance employees' resilience against phishing, smishing, and vishing attacks. These campaigns mimic real-world scenarios, allowing the organization to assess the effectiveness of its training programs and identify areas for improvement.

The results of these campaigns are analyzed, and targeted feedback is provided to employees to reinforce positive behavior and address any vulnerabilities. GitHub uses a variety of scenarios to keep the campaigns challenging and reflective of evolving threat landscapes.

**Question 4: Does everyone in your organization understand the risk associated with cybersecurity, the common ploys used by threat actors, and how to report any suspicious activities for further investigation?**

GitHub fosters a culture of cybersecurity awareness, ensuring that every employee understands the potential risks and common tactics employed by threat actors. Regular communication channels, such as newsletters and internal forums, are utilized to share information about emerging threats and recent incidents.

Employees are encouraged to report any suspicious activities promptly. GitHub has established clear and accessible reporting mechanisms, including an incident response team, to investigate and address reported incidents. Anonymous reporting options are also available to promote a culture of transparency and accountability.

1. **Access Control**

**Question 5: Are all vendor default accounts changed or disabled?**

GitHub follows a strict policy regarding vendor default accounts. Upon integration or deployment of any vendor solutions, default accounts are promptly either changed to unique credentials or disabled if not required. GitHub's security team ensures that any default accounts are properly managed to prevent unauthorized access and minimize potential vulnerabilities.

Question 6: Are only necessary services, protocols, daemons, and functions enabled?

GitHub employs a principle of least privilege in its system configuration. Unnecessary services, protocols, daemons, and functions are disabled by default. System administrators regularly review and update the configuration to ensure that only essential components are active. This practice reduces the attack surface and enhances overall system security.

**Question 7: Is all unnecessary functionality removed or disabled?**

GitHub actively removes or disables any unnecessary functionality across its infrastructure. Reducing the attack surface is a priority, and as part of routine security reviews, any features or services that are deemed non-essential are either removed or disabled. This proactive approach helps to mitigate potential security risks associated with unused or unnecessary functionalities.

**Question 8: Are all accounts immediately disabled or deleted upon termination of employment?**

GitHub has a well-defined account management process. When an employee's employment is terminated, their accounts are immediately disabled to prevent any unauthorized access. The security team ensures that access is promptly revoked across all systems and services. Depending on the organization's policies, accounts may be scheduled for deletion after a certain grace period.

**Question 9: Are all screen idle times set for 15 minutes, and do they require reauthentication to unlock?**

GitHub enforces a security policy where screen idle times are set to a maximum of 15 minutes for workstations and other devices. After this period, systems require reauthentication to unlock, adding an additional layer of protection against unauthorized access in case a user steps away from their workstation. This practice aligns with best security practices to prevent unauthorized access due to unattended devices.

In conclusion, GitHub demonstrates a commitment to robust access control measures, including the management of vendor default accounts, adherence to the principle of least privilege, removal of unnecessary functionality, prompt account management upon employee termination, and the enforcement of screen idle times with reauthentication requirements. These practices contribute to a secure access control environment within the organization.

**Question 10: Do you provide end users a tool to save all passwords (preferably cloud-based for home and work use)?**

GitHub recognizes the importance of secure password management. While the organization encourages the use of password managers, it does not provide a specific tool for end users to save all passwords. Instead, GitHub recommends reputable third-party password management solutions and educates users on the benefits of using such tools for securely storing and generating complex passwords. GitHub ensures compatibility and integration with various password management applications to facilitate user adoption and enhance overall security.

**Question 11: Have you developed an administrator (admin) and user password or passphrase policy that eliminates the use of common or easy-to-guess passwords?**

GitHub has implemented a comprehensive password and passphrase policy for both administrators and users. The policy includes guidelines that prohibit the use of common or easily guessable passwords. GitHub encourages the use of strong, complex passwords or passphrases that meet specific criteria for length, complexity, and uniqueness.

Regular communication and training programs reinforce the importance of adhering to the password policy. GitHub employs mechanisms, such as password complexity requirements and regular password expiration, to ensure that user accounts maintain a high level of security. The organization also encourages the use of multi-factor authentication (MFA) to enhance overall account protection.

1. **End Points**

**Question 12: Are all endpoint logs being ingested by a smart technology that uses threat intelligence and artificial intelligence (AI) based on threat actor activities and heuristics?**

GitHub employs an advanced endpoint security solution that includes a smart technology capable of ingesting endpoint logs. This technology leverages threat intelligence and AI-driven algorithms to analyze and detect potential security threats. The system continuously evolves based on threat actor activities and heuristics, enhancing GitHub's ability to identify and respond to emerging security risks in real-time.

**Question 13: Do you harden all endpoints and remove everything that is not needed for job functionality?**

GitHub follows a rigorous endpoint hardening process. Endpoints are configured to adhere to the principle of least privilege, removing unnecessary components and functionalities that are not essential for job functionality. Regular security audits and reviews ensure that endpoints are hardened to minimize attack surfaces and enhance overall endpoint security.

**Question 14: Do you have next-generation anti-malware protection (e.g., managed detection and response [MDR], extended detection and response [XDR], endpoint detection and response [EDR]) on all endpoints that utilize a threat intelligence-based security analytics platform with built-in security context?**

GitHub has implemented next-generation anti-malware protection on all endpoints. This includes features such as managed detection and response (MDR), extended detection and response (XDR), and endpoint detection and response (EDR). These solutions leverage a threat intelligence-based security analytics platform with built-in security context, enhancing the organization's ability to detect, respond to, and mitigate advanced threats.

**Question 15: Do you prevent non-enterprise-controlled and secured devices from connecting to any portion of your network?**

GitHub enforces strict access controls to prevent non-enterprise-controlled and unsecured devices from connecting to any part of the network. This is achieved through network segmentation, strong authentication mechanisms, and regular monitoring of connected devices. Unauthorized or non-compliant devices are promptly identified and prevented from accessing GitHub's network.

**Question 16: Do all endpoints have personal firewalls for accessing the Internet when not attached to the enterprise network?**

All endpoints at GitHub are equipped with personal firewalls, especially when accessing the Internet outside the enterprise network. These firewalls provide an additional layer of protection, helping to secure endpoints against potential threats and unauthorized access, particularly in external or untrusted network environments.

**Question 17: Do all endpoints have antivirus software installed that cannot be disabled and is automatically updated when new updates are available?**

GitHub ensures that all endpoints have antivirus software installed, and the security settings are configured to prevent users from disabling the protection. Automatic updates are enabled to ensure that the antivirus software stays current with the latest threat definitions and security patches.

**Question 18: Do all endpoints have a next-generation anti-malware application installed?**

GitHub has deployed next-generation anti-malware applications on all endpoints. These applications use advanced techniques and threat intelligence to proactively identify and mitigate malware threats. Regular updates and continuous monitoring contribute to the effectiveness of the anti-malware solution.

1. **Event Management**

**Question 19: Are all logs stored for at least 2 years?**

GitHub adheres to a log retention policy that ensures all logs are stored for a minimum of 2 years. This practice aligns with industry best practices and compliance requirements, allowing for historical analysis, forensic investigations, and regulatory compliance.

**Question 20: Are all devices generating logs?**

GitHub has implemented a comprehensive logging strategy where logs are generated on all devices across its infrastructure. This includes servers, workstations, network devices, and other relevant endpoints. Centralized logging mechanisms are employed to aggregate and manage logs efficiently.

**Question 21: Are all logs being reviewed daily by inside and/or outside sources?**

GitHub conducts daily log reviews, leveraging both internal resources and, when applicable, external sources. This proactive approach ensures prompt detection of any anomalous activities or security incidents. The review process involves analyzing logs for signs of potential threats, unauthorized access, or other security-related events.

**Question 22: Do you have a mature and well-organized cybersecurity incident response (in-house or in conjunction with third parties) that thoroughly investigates all incidents?**

GitHub has established a mature and well-organized cybersecurity incident response framework. The incident response team, whether in-house or in collaboration with third parties, is trained and equipped to thoroughly investigate all incidents. The process includes identification, containment, eradication, recovery, and lessons learned phases.

GitHub's incident response plan is regularly tested through simulations and drills to ensure effectiveness and readiness. Post-incident reviews contribute to continuous improvement, refining response procedures based on lessons learned from each incident.

1. **Security Architecture**

**Question 23: Do you only give employees the tools and access needed to perform their job functions, and nothing else?**

GitHub prioritizes the principle of least privilege, ensuring that employees receive access and tools strictly aligned with their job responsibilities. This practice minimizes the risk of unauthorized access and potential security breaches by limiting user permissions to the essentials required for their tasks.

**Question 24: Do you utilize the principle of least privilege?**

GitHub is committed to the principle of least privilege throughout its organizational structure. By adopting this approach, GitHub ensures that each user, system, or application operates with the minimum level of access necessary for their roles. This proactive measure significantly reduces the attack surface and mitigates the impact of potential security incidents.

**Question 25: Do you deploy a zero-trust model?**

GitHub embraces a zero-trust security model, a paradigm that assumes no implicit trust within the network, regardless of location or user. This model aligns with GitHub's commitment to robust security, requiring continuous verification for every user and device attempting to connect to its systems. This approach enhances overall security posture by eliminating assumptions and implementing stringent authentication and authorization protocols.

**Question 26: Do you require multifactor authentication (MFA) for all connections outside of the network?**

Yes, GitHub enforces multifactor authentication (MFA) for all external connections, bolstering the authentication process. By adding an extra layer of verification beyond traditional passwords, GitHub enhances its defense against unauthorized access attempts, reducing the risk of compromised accounts and potential data breaches.

**Question 27: Do you require MFA for internal authenticated network users to access key infrastructure and data inside the network (i.e., the crown jewels)?**

Absolutely, GitHub mandates multifactor authentication (MFA) for internal network users, particularly when accessing critical infrastructure and sensitive data often referred to as the "crown jewels." This stringent security measure ensures that even within the trusted network, an additional layer of authentication is in place, safeguarding the organization's most valuable assets.

**Question 28: Do you manage all credentials in an order that allows you to quickly conduct a password reset for every account on your network? (This includes service accounts.)**

GitHub has implemented robust credential management practices that enable the organization to promptly conduct password resets for all accounts, including service accounts. This proactive approach is crucial for responding swiftly to potential security incidents, minimizing downtime, and maintaining the integrity of the network.

**Question 29: Have you recently assessed your Active Directory to ensure that it is properly configured and secured?**

GitHub conducts regular and comprehensive assessments of its Active Directory infrastructure. These assessments are designed to validate proper configuration, identify potential vulnerabilities, and ensure that security measures are in place and effective. This proactive approach aligns with GitHub's commitment to maintaining a secure and resilient Active Directory environment.

**Question 30: Are you actively monitoring the security of your Active Directory?**

GitHub maintains a proactive security posture by actively monitoring the security of its Active Directory. Continuous monitoring involves real-time analysis of events, anomalies, and potential threats within the Active Directory environment. This ongoing vigilance enables GitHub to swiftly detect and respond to security incidents, ensuring the integrity and confidentiality of the directory's data and functionality. Regular audits and analysis contribute to the continuous improvement of security measures and incident response capabilities.

**Question 31: Do your perimeter firewalls have a deny-all rule unless otherwise authorized?**

GitHub's perimeter firewalls are configured with a default deny-all rule, ensuring that no traffic is allowed unless explicitly authorized. This approach aligns with the principle of least privilege, significantly reducing the attack surface and mitigating the risk of unauthorized access.

**Question 32: Is your demilitarized zone (DMZ) secured?**

GitHub places a high emphasis on securing its demilitarized zone (DMZ). The DMZ is meticulously designed to act as a buffer between the internal network and external entities. It incorporates robust security controls, such as intrusion detection and prevention systems, to fortify the network perimeter and protect critical assets.

**Question 33: Has it been ensured that there are no data, databases, or stored accounts on the DMZ?**

GitHub maintains a stringent policy to ensure that no sensitive data, databases, or user accounts are stored within the DMZ. The DMZ is treated as a transient space for external-facing services, and data is securely confined to internal, more protected segments of the network to prevent unauthorized access.

**Question 34: Do you deploy anti-spoofing technology to prevent forged IP addresses from entering the network?**

GitHub employs advanced anti-spoofing technologies as part of its network defense strategy. These technologies are designed to detect and block traffic with forged or manipulated IP addresses, enhancing the network's resilience against a range of cyber threats that attempt to exploit address spoofing.

**Question 35: Do you prevent the disclosure of internal IP addresses and routing information on the Internet?**

GitHub has implemented measures to prevent the disclosure of internal IP addresses and routing information on the Internet. This safeguard is crucial for maintaining the confidentiality of the internal network architecture, thwarting potential reconnaissance attempts, and safeguarding against targeted attacks that exploit exposure of such information.

**Question 36: Do you segment key infrastructure from other parts of the network with restrictive firewalls (e.g., segmenting WiFi, confidential data, virtual machines, and printers away from crown jewels)?**

GitHub employs a meticulous network segmentation strategy, leveraging restrictive firewalls to isolate key infrastructure components. This segmentation includes WiFi networks, confidential data repositories, virtual machines, and printers, effectively compartmentalizing these elements away from critical assets often referred to as the "crown jewels." This approach enhances security by minimizing lateral movement and containing potential threats within specific network segments. It also allows for more granular control over access permissions, reducing the risk of unauthorized access to critical systems and data.

1. **Cryptography**

**Question 37: Are procedures defined and implemented to protect cryptographic keys used to protect stored data against disclosure and misuse?**

GitHub has well-defined and implemented procedures to safeguard cryptographic keys used for protecting stored data. These procedures are designed to prevent unauthorized access, disclosure, and misuse of cryptographic keys, ensuring the integrity and confidentiality of sensitive information.

**Question 38: Are cryptographic keys stored in the fewest possible locations with at least dual custodians?**

Indeed, GitHub follows a best practice approach by storing cryptographic keys in the fewest possible locations. Additionally, a dual custodian model is employed, ensuring that access to cryptographic keys requires the involvement of at least two authorized individuals. This dual control enhances security by mitigating the risk associated with a single point of compromise.

**Question 39: Do you utilize full disk encryption on all appropriate drives?**

GitHub prioritizes data security by employing full disk encryption on all appropriate drives. This includes encrypting the entire disk, protecting data at rest and mitigating the risk of unauthorized access in case of physical device theft or compromise.

**Question 40: Do you use secure encryption in motion—at least Transport Layer Security (TLS) 1.1 or higher?**

Absolutely, GitHub employs secure encryption in transit, utilizing at least Transport Layer Security (TLS) 1.1 or higher for data transmission over networks. This ensures the confidentiality and integrity of data during transit, safeguarding it from eavesdropping and potential tampering.

**Question 41: Is all nonconsole administrative access encrypted using strong cryptography?**

GitHub prioritizes the use of strong cryptography for all non-console administrative access. This includes robust encryption mechanisms to protect administrative access channels, ensuring that sensitive administrative activities are conducted securely, and the confidentiality of administrative communications is maintained.

1. **Threats**

**Question 42: Do you perform periodic targeted threat hunts?**

Yes, GitHub conducts regular and periodic targeted threat hunts as part of its proactive security strategy. These hunts involve actively searching for signs of potential threats or malicious activities within the network, ensuring early detection and mitigation of security risks.

**Question 43: Do you ingest current threat intelligence (preferably from more than one source) and have a procedure to implement rapid countermeasures based on good threat intelligence?**

Certainly, GitHub places a strong emphasis on threat intelligence. The organization actively ingests current threat intelligence from multiple reputable sources. A well-defined procedure is in place to implement rapid countermeasures based on this intelligence, enabling swift response to emerging threats and enhancing overall cybersecurity resilience.

**Question 44: Does it include performing routine dark web reconnaissance to learn what exists on the dark web about your brand and enterprise structures?**

Yes, GitHub incorporates routine dark web reconnaissance as part of its threat intelligence practices. This proactive measure helps the organization stay informed about any information on the dark web related to its brand and enterprise structures. This awareness is crucial for preemptive action against potential threats originating from the dark web.

**Question 45: Do you closely monitor all vendor and third-party supply-chain connections for compliance and untoward issues?**

Absolutely, GitHub maintains a vigilant approach to monitoring all vendor and third-party supply-chain connections. This includes continuous monitoring for compliance with security standards and policies. Additionally, the organization actively assesses these connections for any untoward issues, ensuring that third-party relationships do not introduce vulnerabilities or compromise the overall security posture. This practice aligns with GitHub's commitment to a robust and secure supply chain.

1. **Testing**

**Question 46: Do you conduct at least 1 penetration test annually, performed by a third party?**

Yes, GitHub follows a robust security testing regimen, including at least one penetration test annually. These tests are conducted by third-party experts to ensure an objective and comprehensive evaluation of the organization's security posture. This approach provides valuable insights into potential vulnerabilities and helps validate the effectiveness of existing security controls.

**Question 47: Do you conduct routine vulnerability scans and remediate all vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 4 or more within 30 days, and all other vulnerabilities within 90 days?**

GitHub prioritizes routine vulnerability scans to identify and assess potential security weaknesses. Vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 4 or more are remediated within a stringent 30-day timeframe. For other vulnerabilities, the organization ensures remediation within 90 days. This proactive approach minimizes the window of exposure and strengthens the overall security posture.

**Question 48: Do you routinely scan your Internet-facing infrastructure for penetration and vulnerabilities?**

Absolutely, GitHub maintains a proactive stance by routinely scanning its Internet-facing infrastructure for both penetration testing and vulnerability assessments. This regular scrutiny helps identify and address potential weaknesses in the organization's external-facing systems, reducing the risk of unauthorized access and fortifying defenses against evolving cyber threats.

**Question 49: Do you perform an annual business impact analysis/risk analysis report with insider and outside auditors?**

GitHub conducts an annual business impact analysis and risk analysis report, engaging both internal and external auditors. This comprehensive assessment evaluates the potential impact of various risks on the organization's operations. The involvement of insider and outside auditors ensures a thorough and objective analysis, facilitating the identification of key areas for improvement and strategic security enhancements. The outcomes of this analysis contribute to refining the organization's risk management and mitigation strategies.

1. **Policy**

**Question 50: Do you have an enterprise security policy that is updated at least annually and understood by all parties to which it applies?**

Indeed, GitHub maintains a comprehensive enterprise security policy that undergoes regular updates, ensuring its relevance and alignment with evolving security needs. The policy is not merely a static document; it is a living framework that adapts to emerging threats and industry best practices. Moreover, GitHub ensures that all relevant parties, including employees, contractors, and stakeholders, understand the policy through regular communication, training programs, and awareness campaigns. This commitment to communication and education is fundamental to fostering a security-conscious culture within the organization.

**Question 51: Do you have a formal change control policy?**

Yes, GitHub has implemented a formal change control policy as a crucial component of its overall security governance. This policy defines structured processes and procedures for managing changes to the organization's IT infrastructure, systems, and applications. It includes steps for assessing the impact of changes, obtaining necessary approvals, and ensuring that changes are implemented in a controlled and secure manner. The change control policy is designed to prevent unauthorized alterations, reduce the risk of disruptions, and maintain the integrity and availability of systems and services. Regular audits and reviews are conducted to validate adherence to this policy and identify opportunities for continuous improvement.

1. **Physical**

**Question 52: Are processes and mechanisms for restricting physical access to servers, consoles, backup, and network equipment in place and properly safeguarded?**

GitHub has implemented robust processes and mechanisms to restrict physical access to critical infrastructure components such as servers, consoles, backup systems, and network equipment. Physical access controls include secure entry points, biometric authentication, key card systems, and surveillance. These measures ensure that only authorized personnel can access sensitive equipment. Regular audits and monitoring further validate the effectiveness of these controls, contributing to the overall security of the physical infrastructure.

**Question 53: Are physical and/or logical controls implemented to restrict the use of publicly accessible network jacks within the facilities?**

GitHub has implemented a combination of physical and logical controls to restrict the use of publicly accessible network jacks within its facilities. Physical controls include the placement of network jacks in controlled access areas, limiting visibility and accessibility. Logical controls involve network segmentation and access policies to prevent unauthorized devices from connecting to these jacks. This dual-layered approach enhances the security posture, minimizing the risk of unauthorized access or tampering with the network infrastructure within the facilities. Regular monitoring and assessments are conducted to ensure the continued effectiveness of these controls.

1. **Plans**

**Question 54: Do you have a good cyber incident response plan (CIRP) that is reviewed and practiced yearly? The CIRP should be routinely updated, and the core and extended incident response teams should practice responses at least annually using tabletop or functional cybersecurity exercises.**

GitHub maintains a robust Cyber Incident Response Plan (CIRP) that undergoes regular reviews and annual practices. The plan is designed to address the dynamic nature of cybersecurity threats, and updates are made to reflect the evolving threat landscape and organizational changes. The core and extended incident response teams participate in routine tabletop or functional cybersecurity exercises, simulating various scenarios to ensure preparedness and effectiveness in responding to incidents. This proactive approach not only enhances the organization's resilience but also fosters continuous improvement through lessons learned during each exercise.

**Question 55: Do you have playbooks with technical instructions for handling common cybersecurity incidents?**

GitHub has developed comprehensive playbooks containing detailed technical instructions for handling common cybersecurity incidents. These playbooks serve as a valuable resource for incident response teams, providing step-by-step guidance on identifying, containing, eradicating, recovering from, and analyzing security incidents. Each playbook is tailored to specific incident types, ensuring that the response is both swift and effective. Regular reviews and updates to these playbooks align with the organization's commitment to staying ahead of emerging threats and evolving best practices in incident response.

1. **Inventory**

**Question 56: Do you have thorough diagrams of the entire network, including WiFi?**

GitHub maintains comprehensive network diagrams that encompass the entire infrastructure, including WiFi networks. These diagrams provide a detailed visualization of the network architecture, mapping out the relationships between different components, and highlighting the connectivity of WiFi networks. These visual aids serve as valuable references for network management, troubleshooting, and security assessments.

**Question 57: Do you have a complete inventory of all assets that includes business criticality levels, owners, co-owners, and restoration? Does this inventory include instructions with time periods to recover?**

GitHub maintains a meticulous inventory of all assets, ensuring a detailed record that includes business criticality levels, ownership information, co-owners, and restoration procedures. This comprehensive inventory serves as a foundation for effective asset management and risk assessment. Moreover, each asset in the inventory is associated with clear instructions for recovery, including specified time periods for restoration. This approach aligns with GitHub's commitment to resilience and efficient response in the face of disruptions.

**Question 58: Do you have a full set of data flow diagrams?**

Indeed, GitHub has developed a complete set of data flow diagrams that depict the movement of information within its systems and networks. These diagrams provide a detailed representation of how data is processed, stored, and transmitted across various components of the infrastructure. This visual representation aids in understanding the flow of data, identifying potential points of vulnerability, and enhancing overall data security. Regular reviews and updates to these data flow diagrams contribute to maintaining an accurate and up-to-date understanding of the organization's information architecture.

1. **Data Management**

**Question 59: Do you utilize file integrity monitoring (FIM) of the crown jewels of the organization?**

Certainly, GitHub employs file integrity monitoring (FIM) specifically focused on the crown jewels of the organization. This involves continuous monitoring and validation of critical files and systems to detect any unauthorized changes. FIM ensures the integrity and security of these crown jewel assets by promptly identifying and responding to potential security incidents or breaches.

**Question 60: Is storage of confidential data kept to a minimum and securely deleted after it’s no longer needed?**

GitHub follows a strict data management policy to minimize the storage of confidential data to the essential necessities. Additionally, confidential data is securely deleted in accordance with predefined protocols once it is no longer required. This practice not only reduces the risk of unauthorized access but also aligns with privacy and compliance standards.

**Question 61: Do you require data classification throughout the network?**

Yes, GitHub enforces a comprehensive data classification policy across its network. This policy ensures that data is categorized based on its sensitivity and criticality, allowing for the implementation of appropriate security controls and access restrictions. Data classification is fundamental to effective data protection, providing a structured approach to safeguarding information assets.

**Question 62: Do you deploy a network and cloud-based data loss prevention (DLP) program wherever confidential data resides?**

Answer: Absolutely, GitHub employs a robust data loss prevention (DLP) program that spans both network and cloud environments. This program is strategically implemented wherever confidential data resides, actively monitoring and preventing unauthorized access, transmission, or exfiltration of sensitive information. This approach aligns with GitHub's commitment to maintaining the confidentiality and integrity of its data assets.

**Question 63: Do you prevent confidential data from being copied to external devices and external devices from being attached to endpoints?**

Yes, GitHub has implemented measures to prevent the unauthorized copying of confidential data to external devices and restricts the attachment of external devices to endpoints. This proactive security measure helps mitigate the risk of data leakage and ensures that confidential information remains within controlled environments. The organization enforces policies and utilizes technical controls to reinforce this protection against potential insider and external threats.

1. **Software Development**

**Question 64: Are processes and mechanisms for developing and maintaining secure systems and software defined and understood?**

GitHub has well-defined processes and mechanisms for developing and maintaining secure systems and software. These processes are thoroughly documented and widely understood across the software development teams. This ensures a consistent and unified approach to secure coding practices throughout the development lifecycle.

**Question 65: Are software engineering techniques or other methods defined and in use by software development personnel to prevent or mitigate common software attacks and related vulnerabilities in all software?**

GitHub integrates software engineering techniques and other methods to prevent and mitigate common software attacks and vulnerabilities. Secure coding practices are embedded into the development process, and developers receive training on identifying and addressing potential security risks. This proactive approach helps reduce the likelihood of introducing vulnerabilities and enhances the overall security posture of the software.

**Question 66: With regard to public-facing web applications, are new threats and vulnerabilities addressed on an ongoing basis?**

GitHub is committed to addressing new threats and vulnerabilities in public-facing web applications on an ongoing basis. Regular security assessments, code reviews, and continuous monitoring are part of the development lifecycle. This approach enables the identification and prompt remediation of emerging threats, ensuring that the web applications remain resilient against evolving security challenges.

**Question 67: Are these applications protected against attacks?**

Yes, GitHub employs robust security measures to protect public-facing web applications against a variety of attacks. These measures include web application firewalls, intrusion detection systems, secure coding practices, and ongoing monitoring. The organization prioritizes a defense-in-depth strategy to safeguard these applications and their underlying infrastructure.

**Question 68: Are preproduction environments separated from production environments, and is separation enforced with access controls?**

GitHub maintains a clear separation between preproduction and production environments, and this separation is strictly enforced through robust access controls. Access to preproduction environments is restricted to authorized personnel, and stringent controls are in place to prevent unauthorized access or data leakage. This separation ensures that testing and development activities do not impact the integrity and security of the production environment.

1. **Mobile Devices**

**Question 69: Are all mobile devices governed by effective mobile device management (MDM) policies?**

GitHub enforces comprehensive mobile device management (MDM) policies to govern all mobile devices used within the organization. These policies are designed to ensure that mobile devices adhere to security standards, configurations, and access controls. MDM plays a crucial role in maintaining the integrity and security of mobile devices by enforcing encryption, setting authentication requirements, and enabling remote device management capabilities.

**Question 70: Do you disallow any connectivity of mobile devices not controlled by enterprise security mechanisms?**

GitHub strictly disallows any connectivity of mobile devices that are not controlled by enterprise security mechanisms. This proactive measure ensures that only devices adhering to the organization's security standards and policies can connect to enterprise networks or access sensitive data. By preventing connections from unmanaged devices, GitHub mitigates the risk of unauthorized access, data breaches, and potential compromise of enterprise resources. This approach aligns with the organization's commitment to maintaining a secure and controlled mobile device environment.

**References**

<https://github.com/security>

<https://resources.github.com/contact/security/>

<https://securitylab.github.com/research/>

[The Effectiveness of Security Interventions on GitHub](https://arxiv.org/pdf/2309.04833.pdf)

[Asleep at the Keyboard? Assessing the Security of GitHub Copilot’s Code Contributions](https://ieeexplore.ieee.org/document/9833571)

[Everything You Need To Know About GitHub Security](https://www.legitsecurity.com/github-security-best-practices)