Jeytha Sahana Venkatesh Babu

Project Report : OSINT-Based Cybersecurity Risk Assessment: Netflix

Pace University

Cyber Analysis and Modelling - CYB 651

Introduction

This paper profiles Netflix, Inc., leveraging open-source intelligence (OSINT) to assess its publicly visible network infrastructure, key technology platforms, and potential cybersecurity vulnerabilities. Passive data was gathered using tools such as Robtex, MXToolbox, Exploit Database (Exploit-DB), and the Cybersecurity and Infrastructure Security Agency's (CISA) Known Exploited Vulnerabilities (KEV) catalog. The assessment will identify Netflix's significant reliance on cloud providers (primarily Amazon Web Services), its extensive microservices architecture utilizing open-source tools like Genie, Apache Kafka, and Cassandra, and potential weaknesses such as subdomain sprawl or email authentication gaps. The report will contextualize Netflix's threat landscape by examining past security incidents, including the critical Netflix Genie bug (CVE-2024-4701) and cloud misconfigurations, alongside broader industry-specific threats like credential stuffing and API abuse. Proposed mitigations will include strengthening cloud security posture, enhancing API and open-source software security, and optimizing identity and access management.

Methodology

- Robtex was used for DNS recon to identify hosts and subdomains.
- MXToolbox was used to assess email security policies like DMARC.
- CISA KEV Catalog was consulted to find actively exploited vulnerabilities in technologies used by Netflix.
- News and Regulatory Reports were reviewed for historical data breaches and industry security trends.
- Public Company Filings & Blogs provided insights into Netflix's technology stack and security practices.

Company Overview

Netflix is a tech company at its core, not just a media company. The streaming service itself is built on technology. It was one of the first companies to widely use **cloud computing**, primarily with Amazon Web Services (**AWS**), and its system is built using a **microservices architecture**. This allows it to easily scale and deliver content reliably to people all over the world on various devices. This foundation also helps them innovate quickly and roll out new features.

Because of the valuable assets it holds, Netflix is a major target for cyberattacks. These assets include:

- **Customer data**: This includes personal information, what people watch, and their payment details.
- **Intellectual property**: This covers their original shows and movies, their special recommendation system, and production secrets.
- **Service availability**: It is crucial for Netflix to always be available. Any major outage or security breach could directly hurt its revenue, damage its brand, and cause customers to lose trust.

Protecting these assets is a top priority for the company.

Findings: Network and Infrastructure

Cloud-Native Primary Platform: Primarily uses **Amazon Web Services (AWS)** for hosting services (EC2, S3, Lambda, VPC, IAM).

Content Delivery Network (CDN): Leverages proprietary **Open Connect CDN** complemented by **cloud-native WAFs** (e.g., AWS WAF) or third-party solutions for content delivery and attack mitigation.

Extensive Subdomain Sprawl: Features numerous public subdomains (e.g., api.netflix.com, dev.netflix.com) indicating a large, potentially exposed attack surface.

Complex Cloud Architecture: Inferred as a heavily segmented, cloud-based system with **internal DNS resolution** and **service mesh** for microservices communication.

Findings: Email Security Posture

The MXToolbox analysis of netflix.com provided critical insights into their external email security posture, which is foundational for defending against phishing and impersonation attacks.

- **Email Routing (MX Records):** Netflix uses a **third-party email security provider** to filter all inbound email traffic, relying on their capabilities for initial defense.
- **SPF Record Analysis:** A comprehensive **SPF record** is in place, but its inclusion of multiple third-party senders introduces a **supply chain risk**.
- **DKIM Record Analysis: DKIM records are correctly configured**, ensuring Netflix digitally signs outgoing emails for authenticity and integrity verification.
- **DMARC Policy Analysis: A Critical Finding:** The p=none **DMARC policy** renders the domain highly **vulnerable to spoofing and phishing attacks**, as emails failing authentication are not quarantined or rejected.

Technology Platforms

Netflix's public-facing services and internal operational security posture are intrinsically linked to its choice and implementation of cutting-edge technology platforms.

- **Primary Cloud Platform (Amazon Web Services - AWS):** Netflix heavily relies on AWS, requiring them to manage security *in* the cloud for services like EC2, S3, Lambda, and IAM, as per the shared responsibility model.
- **Microservices Architecture:** Netflix's pioneering microservices architecture provides agility and scalability but introduces complex security challenges in inter-service communication and policy management across many independent services.

- **Containerization and Orchestration:** Netflix uses Docker and its proprietary Titus orchestration platform on AWS, demanding robust security for container image management, runtime protection, and hardening of the orchestration plane due to dynamic scaling.
- **Data Storage and Analytics (Apache Cassandra, Kafka, Hadoop/Spark Ecosystem):** Netflix's distributed data ecosystem relies on Cassandra, Kafka, and the Hadoop/Spark ecosystem, making data security at rest and in transit, and continuous vulnerability management for these core systems, critical.
- **Internal Security Tools/Frameworks:** Netflix develops internal security tools like Security Monkey and Spinnaker, whose security is crucial as their compromise could grant deep access to core infrastructure, while open-source contributions like Genie also introduce potential exposure.

Common Threats to Streaming/Media Sector:

- **Credential Stuffing/Account Takeovers:** This remains a pervasive and significant threat. Threat actors commonly use leaked username and password combinations obtained from other data breaches and attempt to "stuff" them against Netflix accounts.
- **Content Piracy/Digital Rights Management (DRM) Bypass:** As a content creator and distributor, Netflix is in an ongoing battle against illegal content distribution. Adversaries constantly seek to bypass Digital Rights Management (DRM) protections to pirate movies and shows, which directly impacts Netflix's revenue and the value of its intellectual property. Security measures around content protection and delivery are paramount.
- **API Abuse:** Netflix operates a vast number of APIs to support its various applications (web, mobile, TV apps) and microservices. Malicious actors may attempt to exploit these APIs for unauthorized data access, service disruption or to manipulate platform features for nefarious purposes.
- **Phishing Campaigns:** Due to its massive global subscriber base, Netflix users are a frequent and lucrative target for highly sophisticated phishing emails and messages

- **Distributed Denial-of-Service (DDoS) Attacks:** The threat of large-scale DDoS attacks remains constant. Such attacks, aimed at overwhelming Netflix's infrastructure, could disrupt service availability, prevent users from streaming content, and cause significant reputational and financial damage.

Netflix's Past Security Incidents: While Netflix has maintained a robust security reputation and has not suffered a widely publicized, large-scale data breach of its core subscriber database various smaller incidents and specific vulnerabilities are still relevant and inform its threat landscape:

- **Supply Chain Vulnerability (Post-Production Company Breach):** In a notable incident, a post-production company working with Netflix experienced a breach that led to the theft and leak of unaired episodes of "Orange Is the New Black" prior to its official release. This highlights the critical importance of supply chain security and third-party risk management for a company that relies heavily on external partners for content creation and distribution.
- **Persistent Credential Stuffing Attacks:** As mentioned, credential stuffing attacks are a well-documented and persistent issue that Netflix and its users continuously face. Netflix has publicly acknowledged these efforts and implements various countermeasures, but the ongoing nature of these attacks demonstrates a continuous risk to user accounts.
- **Specific Bug Bounty Program Disclosures:** While not always leading to public news headlines, Netflix's active bug bounty program has seen disclosures of various vulnerabilities. These often lead to private remediation but indicate the ongoing effort required to secure a platform of its complexity.
- **Regulatory Scrutiny:** Netflix has faced regulatory scrutiny and, in some cases, fines related to data privacy and security practices, particularly concerning international data protection laws. For instance, reports indicate a fine from the South Korean government for allegedly collecting user information without proper consent, underscoring the critical importance of not just technical

security measures, but also adherence to evolving legal and privacy compliance frameworks globally.

Known Exploited Vulnerabilities

Critical Netflix Genie Bug (CVE-2024-4701): The most notable and recently disclosed vulnerability directly related to Netflix's open-source tooling is the critical remote code execution (RCE) bug, CVE-2024-4701. This severe vulnerability, with a near-maximum CVSS score of 9.9, impacts the open-source version of Netflix's Genie job orchestration engine. While Netflix actively uses Genie internally, the immediate impact and remediation efforts publicly focused on external organizations running their own instances of the open-source software.

- **Vulnerability Type:** The core of CVE-2024-4701 is a path traversal bug (CWE-22). This vulnerability arises during the file upload process via a specific Genie API endpoint. A path traversal flaw fundamentally allows an attacker to manipulate file paths to write files to unintended and often restricted locations on the server's file system, effectively "breaking out" of the intended secure directory.
- **Attack Method:** Researchers from Contrast Security were credited with discovering that a specific Genie API, designed to allow users to submit SQL queries by uploading a SQL file, was susceptible to this path traversal attack. An unauthenticated attacker could craft a malicious filename within the upload request to trick the application into uploading the file to a directory outside the designated, restricted upload directory.
- **Impact:** A successful exploitation of this vulnerability could grant an unauthenticated attacker the ability to write arbitrary files to the underlying system. This capability is extremely dangerous as it can directly lead to remote code execution (RCE). With RCE, the attacker gains full control over the compromised server, allowing them to execute arbitrary commands, install malware, or pivot to other systems. More critically for Netflix's context, such control could provide

unauthorized access to the large datasets that Genie is designed to manage and interact with, potentially exposing sensitive subscriber data or intellectual property. The severity is compounded by the fact that the vulnerability requires no special user privileges or interaction from the victim, making it easily exploitable.

- **Remediation:** Netflix rapidly addressed this vulnerability by issuing a fix in Genie OSS version 4.3.18. The company's public security advisory, available on its GitHub repositories, strongly urged all external organizations running open-source Genie instances to upgrade immediately. The advisory also provided specific guidance, noting that organizations not using the local file system to store attachments for Genie were not vulnerable. This incident highlights the dual responsibility of companies like Netflix in managing both their internal deployments and the security of their open-source contributions.

Development & Operations Tooling Vulnerabilities

Netflix is a well-known open-source contributor and user. The information provided lists several open-source tools that are or were part of their stack, including Apache Kafka and Apache Cassandra.

- **Apache Kafka:** The provided information from Apache's official CVE list details several vulnerabilities that could impact Kafka. For example, a vulnerability in Kafka Clients allows attackers to read arbitrary files from the disk and environment variables (CVE-2024-31141). Another vulnerability (CVE-2023-25194) allows for Remote Code Execution (RCE) via a specific configuration. These types of flaws are particularly dangerous in a microservices architecture, as they could allow an attacker to pivot from a single compromised service to a broader part of the network.
- **Apache Cassandra:** The Apache Cassandra project has had its share of vulnerabilities. For instance, a recent vulnerability (CVE-2025-24860) was found to allow users with restricted access to bypass authorization and gain access to data they should not be able to. In a large-scale

database deployment like Netflix's, such a flaw could be an immediate and severe risk, as it could lead to data access or data modification. The existence of public Proof-of-Concept (PoC) code for these vulnerabilities on sites like Exploit-DB would make them a high-priority risk for Netflix.

Cloud Misconfigurations: Public reports, including those derived from Netflix's own bug bounty program disclosures, have repeatedly mentioned past incidents involving misconfigured AWS resources. Examples include S3 buckets with overly permissive public access policies, insecure security group configurations, or misconfigured IAM roles. While these misconfigurations may not always be assigned a formal CVE number, they represent a ongoing risk. Such vulnerabilities can directly lead to unauthorized data access, which shows a major threat to user privacy, intellectual property, and compliance.

Recommended Mitigations

Based on the comprehensive OSINT findings and identified vulnerabilities, the following concrete and actionable cybersecurity mitigations are proposed for Netflix. These recommendations aim to address the identified risks and bolster the company's cyber resilience, particularly within its cloud-native and open-source-reliant environment.

- **Cloud Security:** Continuously scan the AWS environment for misconfigurations and embed security scanning into the CI/CD pipeline for Infrastructure as Code.
- **Identity & Access:** Enforce least privilege, require phishing-resistant MFA, and use Identity-Aware Proxy for secure remote access.
- **API Security:** Deploy robust API gateways and a Zero Trust service mesh for all inter-service communication.
- **Email Security:** Immediately enforce the DMARC p=reject policy to prevent domain spoofing and strengthen employee security training.
- **Vulnerability Management:** Prioritize patching of CISA KEV vulnerabilities, automate container scanning, and secure the open-source software supply chain.

- **Network & Incident Response:** Maintain fine-grained network segmentation and conduct regular cloud-specific incident response drills.
- **Data Protection:** Ensure all sensitive data is encrypted at rest and in transit and implement Data Loss Prevention (DLP).

Conclusion

This OSINT-based cybersecurity risk assessment of Netflix has demonstrated that while the company maintains an exceptionally robust and sophisticated security posture, underpinned by its deep cloud-native expertise and proactive security culture, the inherent scale, distributed complexity, and rapid evolution of its environment, combined with the continuous ingenuity of cyber threats, present ongoing and significant challenges. Crucially, the assessment highlighted the critical importance of security in open-source components with the discovery of the Critical Netflix Genie Bug (CVE-2024-4701), an RCE vulnerability in an internally used and externally distributed tool. This, coupled with the persistent risk of cloud misconfigurations (like overly permissive S3 buckets) and known vulnerabilities in core distributed systems like Apache Kafka and Cassandra, underscores that even the most advanced organizations face continuous threats from fundamental security flaws.

For a company of Netflix's global stature, continuous, proactive security measures that span proprietary systems, cloud infrastructure, and open-source contributions are not merely best practice—they are imperative to protect its vast user base, invaluable content, and global brand reputation in an ever-evolving and highly aggressive digital landscape.

References

- **Netflix. (2025). *Netflix TechBlog*. Netflix TechBlog. https://netflixtechblog.com/**
- **Amazon Web Services (AWS). (n.d.). Netflix on AWS.**

  https://aws.amazon.com/solutions/case-studies/netflix/
- **CISA. (n.d.). Known Exploited Vulnerabilities Catalog**

  .https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- **National Institute of Standards and Technology (NIST). (n.d.).**

  https://nvd.nist.gov/vuln/detail/CVE-2024-4701
- **HackerOne. (n.d.). Netflix Bug Bounty Program.**

  https://hackerone.com/netflix
- ***NVD - CVE-2024-4701*. (2024). Nist.gov. https://nvd.nist.gov/vuln/detail/CVE-2024-4701**
- **National Institute of Standards and Technology (NIST). (n.d.). National Vulnerability Database (NVD).**
  https://nvd.nist.gov/vuln/detail/cve-2023-25194
- **Apache Software Foundation. (n.d.). Apache Kafka Vulnerabilities**
  https://kafka.apache.org/cve-list.html
- **Apache Software Foundation. (n.d.). Apache Cassandra Vulnerabilities.**

  https://cassandra.apache.org/security
- **National Institute of Standards and Technology (NIST). (n.d.). National Vulnerability Database (NVD).**

  https://nvd.nist.gov/vuln/detail/CVE-2025-24860
- **MXToolbox. (n.d.). DMARC Record Lookup.**

  https://mxtoolbox.com/dmarc.aspx
- **CISA. (2025). *Known Exploited Vulnerabilities Catalog*.**

  https://www.cisa.gov/known-exploited-vulnerabilities-catalog

- **Amazon Web Services (AWS). (n.d.). The AWS Shared Responsibility Model.**

  https://aws.amazon.com/compliance/shared-responsibility-model/

- **Cloud Security Alliance. (2017). Cloud Security Architecture.**

  https://cloudsecurityalliance.org/artifacts/cloud-security-architecture/

- **CISA. (n.d.). Known Exploited Vulnerabilities Catalog.**

  https://www.cisa.gov/known-exploited-vulnerabilities-catalog

- **CISA. (2023). Implementing Phishing-Resistant MFA.**

  https://www.cisa.gov/resources-tools/resources/implementing-phishing-resistant-mfa

- **Google. (n.d.). BeyondCorp: A Zero Trust Approach to Security.**

  https://cloud.google.com/beyondcorp

- **National Institute of Standards and Technology (NIST). (2020). Zero Trust Architecture. (NIST Special Publication 800-207).**

  https://doi.org/10.6028/NIST.SP.800-207

- **Open Web Application Security Project (OWASP). (n.d.). API Security Top 10.**

  https://owasp.org/www-project-api-security/