Cyber Attacks and Defenses in Communication Networks

Jeramy Canals, Manuel Capiendo, Tariq Khan, Amaan Ahmed, Noah Steuart, Nicolas Seda

Published: October 2025

Abstract

Communication networks are the foundation of modern society, they allow the secure exchange of information across organizations, governments, and defense systems. But these networks face increasing risks from cyberattacks that exploit both technical flaws and human error. This report analyzes six major attack types: ransomware, zero-day exploits, phishing and spear phishing, SQL injection, DNS spoofing, and man-in-the-middle (MITM), and the related defenses designed to protect communication networks. Each section defines the attack, explains how it works, highlights real-world examples, and describes defense techniques such as patch management, network segmentation, encryption, multi-factor authentication, and user awareness training. The report shows how these attacks threaten data integrity, confidentiality, and availability, within the Department of Defense (DoD) and other critical-infrastructure networks. Overall, the report demonstrates that layered security combining technical controls, user education, and continuous monitoring is essential to maintaining reliable communication networks and national cybersecurity resilience.

Cyber Attacks and Defenses in Communication Networks

Communication networks function as the critical foundation that connects computers, organizations, and global systems. They enable governments, industries, and individuals to exchange information instantly, making them essential to both civilian life and national defense. As dependency on these networks has grown, so have the risks associated with cyberattacks. Every message, transaction, and data transfer relies on the integrity and availability of network infrastructure, yet attackers continuously seek to exploit vulnerabilities in hardware, software, and human behavior.

Network attacks are among the most significant threats facing modern organizations because they target the very systems that enable communication and coordination. The cost of these incidents is overwhelming. Global ransomware damages are projected in the billions annually, and zero-day exploits can silently compromise systems for months before discovery. Phishing remains the most common entry point for breaches, while SQL injections continue to compromise poorly secured web applications. Attacks on the Domain Name System (DNS) and man-in-the-middle (MITM) interceptions show how easily communication channels can be taken over or tampered with. Beyond financial losses, these attacks threaten national security by exposing defense data, disrupting supply chains, and reducing public trust in digital systems.

The Department of Defense (DoD) and allied agencies face particularly high stakes. A single compromise can disrupt mission operations, leak classified information, or disable command-and-control communications. These threats are not hypothetical, the 2021 Colonial Pipeline ransomware incident, the Stuxnet worm, and repeated phishing campaigns against defense contractors highlight the growing sophistication of cyber adversaries. Communication networks

have become battlefields in an ongoing cyber war, where it is often hard to tell the difference between criminals and government-backed attackers.

This project focuses on six major cyberattacks that highlight the diverse methods used to compromise communication networks: ransomware, zero-day exploits, phishing and spear phishing, SQL injection, DNS spoofing, and man-in-the-middle (MITM) attacks. Each section examines the attack's process, notable real-world examples, and effective defense strategies. The goal is to provide a clear understanding of how each attack operates, the weaknesses it exploits, and the layered countermeasures required to defend against it. Together, these analyses demonstrate the urgent need for continuous awareness, proactive security design, and cooperation across organizations to safeguard the integrity of global communication networks.

Cyber Attacks and Defenses

Ransomware

Ransomware is a type of malware that encrypts or locks a victim's data or systems, denying access until a ransom, usually in cryptocurrency, is paid for a decryption key (National Institute of Standards and Technology [NIST], 2023). It is extortion in which attackers use encryption to make data unusable and demand payment for restoration. Ransomware variants often combine encryption with data theft, threatening to leak stolen information to pressure victims (Check Point Software Technologies, 2024). Within the Department of Defense (DoD), ransomware is a serious threat to missions, classified information, and the defense supply chain.

Ransomware attacks usually follow a structured sequence called the kill chain. The first stage, initial access, often occurs through phishing emails, malicious attachments, or unpatched software and misconfigured remote access (CrowdStrike, 2024). Once inside, attackers escalate privileges, spread across systems, and deploy the ransomware. Many ransomware attacks steal sensitive data before encrypting files to strengthen ransom demands (Begovic et al., 2023). When the attack is launched, files are locked and replaced with encrypted versions, and a ransom note demands payment for decryption (Federal Bureau of Investigation [FBI], 2024). Even if the ransom is paid, victims are not guaranteed restoration, and permanent data loss is common.

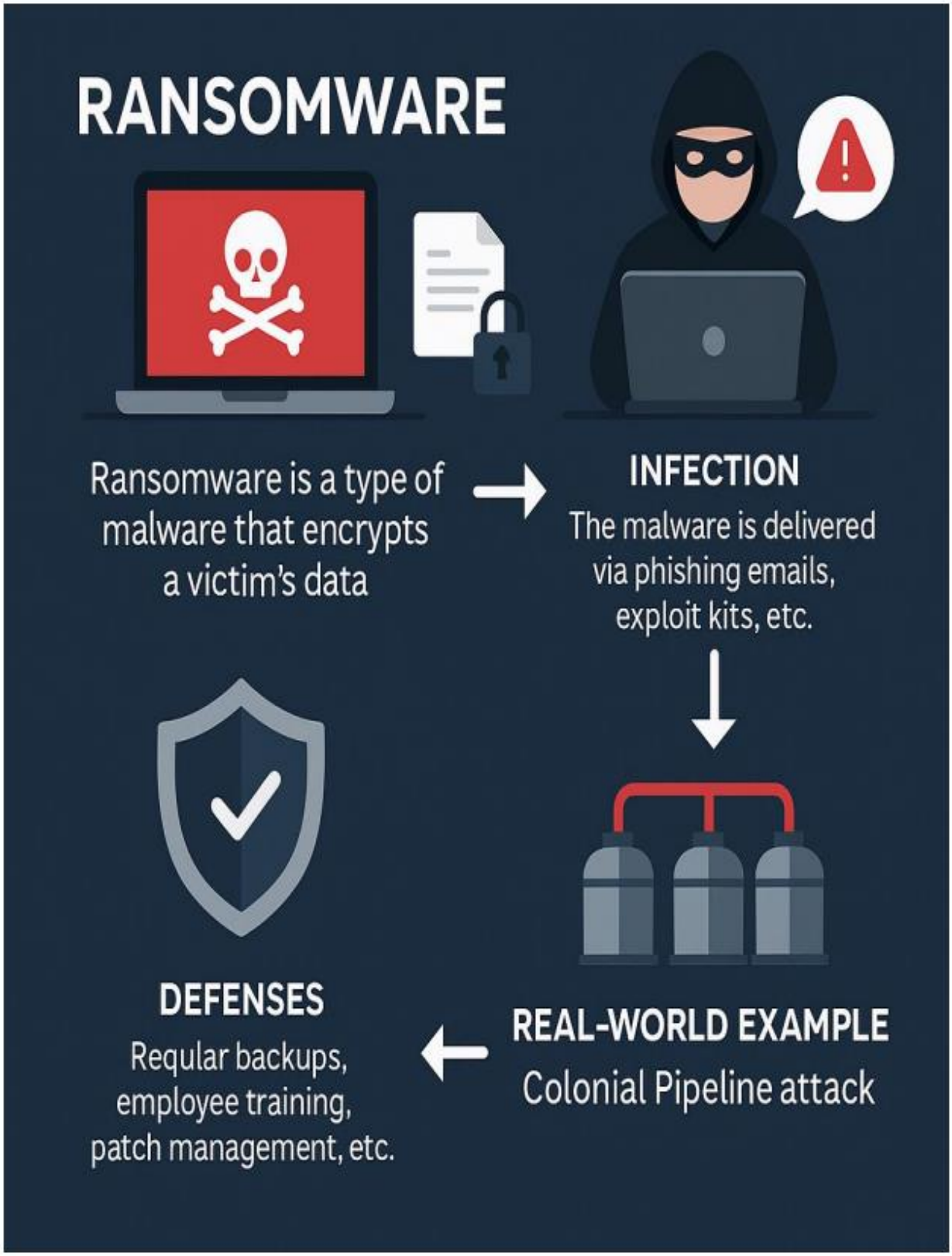
A major example was the 2021 Colonial Pipeline attack by the DarkSide ransomware group. The attack disrupted operations of one of the largest U.S. fuel pipelines, which supplies about 45 percent of the East Coast's fuel (Cybersecurity and Infrastructure Security Agency [CISA], 2021). Colonial Pipeline is a private company, but the attack directly affected national security, causing responses from the DoD and Department of Energy (Department of Homeland Security

[DHS], 2021). The company shut down operations for nearly a week, leading to significant fuel shortages. The attack revealed how vulnerable critical infrastructure is to ransomware and how private-sector breaches can disrupt defense logistics and homeland security.

To defend against ransomware, the DoD and federal agencies focus on Zero Trust Architecture, least privilege, and network segmentation to limit damage if intrusions happen. CISA (2024) recommends offline backups, multi-factor authentication (MFA), and consistent patch management to block common entry points. Behavior-based detection tools can identify weird encryption activity before systems are locked (Begovic et al., 2023). The Cybersecurity Maturity Model Certification (CMMC) makes sure defense contractors maintain strong security controls to protect DoD data.

Even with these defenses, ransomware remains a strong threat. Attackers disable backups, exploit unpatched software, and use social engineering to bypass training. Smaller Defense Industrial Base (DIB) contractors lack the resources for constant monitoring (BlueVoyant, 2024). Paying ransoms can violate U.S. sanction laws, and there is no guarantee of data recovery (FBI, 2024).

Ransomware directly impacts communication networks, the backbone of data exchange between systems and organizations. When ransomware spreads through a network, it disrupts or corrupts the flow of information. In the DoD, disruptions can break secure communication channels, delay logistics, and prevent command systems from transmitting mission-critical information. Understanding network behavior enables cybersecurity teams to detect weird activity early, isolate infections, and protect essential communication systems from collapse.


RANSOMWARE
Ransomware is a type of malware that encrypts a victim's data
INFECTION
The malware is delivered via phishing emails, exploit kits, etc.
REAL-WORLD EXAMPLE
Colonial Pipeline attack
DEFENSES
Regular backups, employee training, patch management, etc.

Zero-Day Exploits

Zero-day vulnerabilities are essentially software security flaws that attackers find before the developers of the software become aware of this exploitable feature. That is where it gets its name “zero-day”, as it means that the developers have literally zero days to deal with the problem as it is already being exploited upon its discovery. This is where there have not been any patches made, so attackers are able to easily exploit and share this vulnerability, putting systems, companies, and users at risk. What is unique about this vulnerability is that they are always unknown until they are being exploited.

To exploit these vulnerabilities, these attackers go through a process that involves first identifying a vulnerability through the code. They will do this through reverse engineering for weaknesses, social engineering with phishing for access into secure systems, or even purchasing vulnerabilities through the black market. Next, they will create custom malware to exploit the discovered flaw and then deliver this payload through means such as phishing, infected USB drives, unsafe websites, or through the network directly.

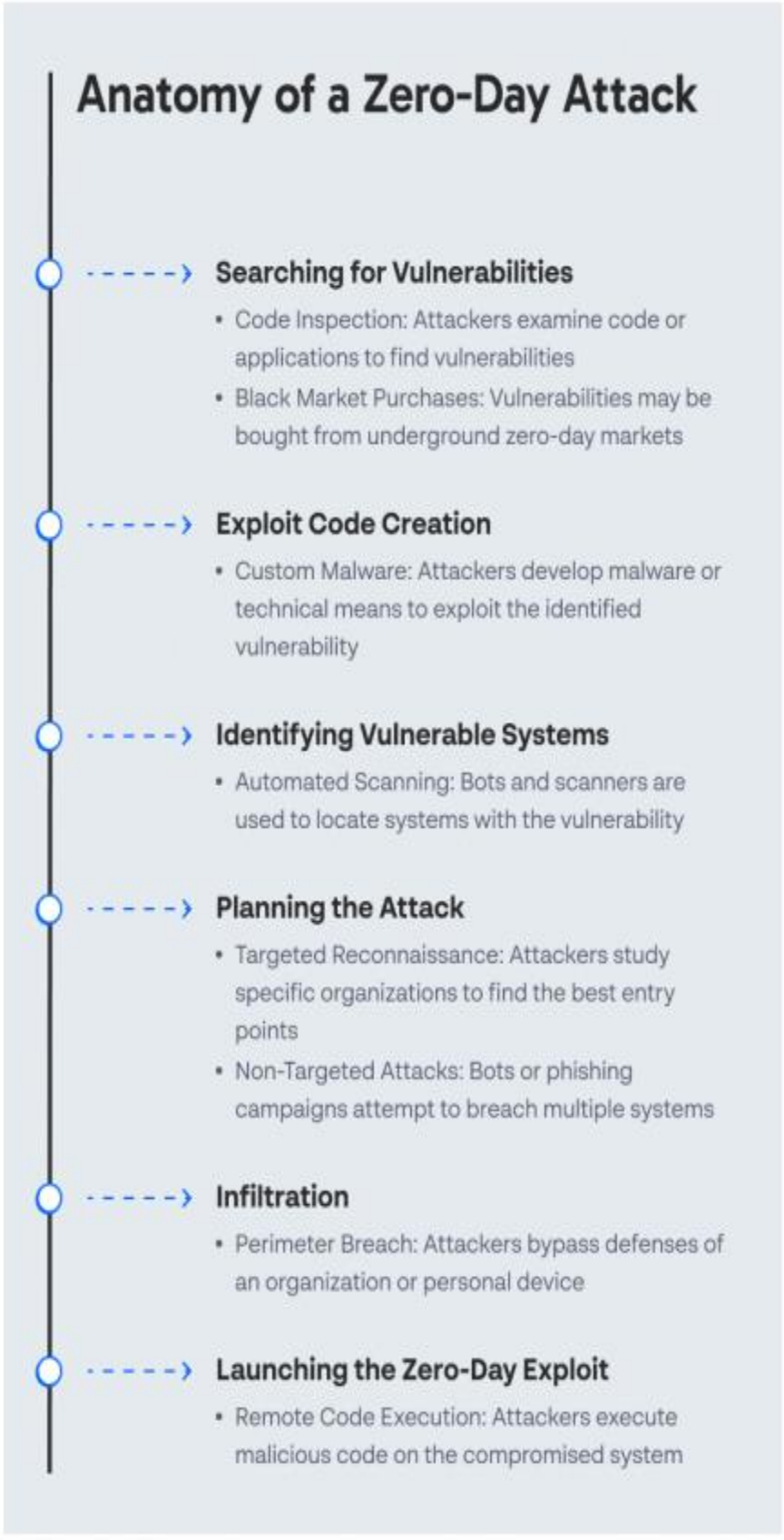
Because of the nature of these exploits the biggest problem that arises to defend against zero-day exploits is the lag between the time the exploit has been abused and when developers identify it and can release a patch. There will always be a lag between this time as developers will not know it even exists until it has been used. This means there is always an advantage for attackers as they have the leverage and time to use this exploit while patches are not developed and systems are not updated.

A particularly good example of this is one of the most famous examples of a zero-day attack that happened in 2010 with a nuclear facility in Iran. This was Stuxnet, a worm that infected an air

gapped nuclear facility through an infected USB that was able to destroy about 1,000 centrifuges through mechanical manipulation, fake data, and be completely undetectable for as long as it has been in the system. This was a Windows attack that was able to go after the programmable logic controllers in these facilities.

Defending against these exploits is difficult as they are naturally found in places that cannot be detected. The main response to this is to create a system of patches to quickly update and address vulnerabilities when they are found. Before this, it is also imperative that there is a system and process to even identify these issues when they do occur and be able to relay that information to users for their safety. Sandboxing is another tool to test suspicious code in a safe environment to detect any malicious code.

Because of such a complex and almost invisible threat, the importance of these attacks in communication networks cannot be overstressed. These networks are the infrastructure of everything that runs in this modern digital age. Energy, media, communication, government, and more rely on these networks to operate foreign adversaries, malicious groups, and anyone can discover and exploit these vulnerabilities. As networks get more complex and upgraded and added onto, it is important to be able to be aware, and detect for the next attack.


Anatomy of a Zero-Day Attack
Searching for Vulnerabilities
• Code Inspection: Attackers examine code or applications to find vulnerabilities
• Black Market Purchases: Vulnerabilities may be bought from underground zero-day markets
Exploit Code Creation
• Custom Malware: Attackers develop malware or technical means to exploit the identified vulnerability
Identifying Vulnerable Systems
• Automated Scanning: Bots and scanners are used to locate systems with the vulnerability
Planning the Attack
• Targeted Reconnaissance: Attackers study specific organizations to find the best entry points
• Non-Targeted Attacks: Bots or phishing campaigns attempt to breach multiple systems
Infiltration
• Perimeter Breach: Attackers bypass defenses of an organization or personal device
Launching the Zero-Day Exploit
• Remote Code Execution: Attackers execute malicious code on the compromised system

Phishing and Spear Phishing

Phishing and spear phishing are among the most common and dangerous cyberattacks, exploiting human trust rather than directly targeting technical systems. This document provides a detailed explanation of what these attacks are, how they work, and the defense mechanisms that can be used to mitigate them.

Phishing is a type of cyberattack in which attackers impersonate a trusted person or organization through emails, fake websites, or messages to trick victims into revealing sensitive information. It relies on social engineering, exploiting human trust rather than directly breaking into computer systems. Example: An email pretending to be from a bank asking you to verify your account by clicking a link.

Spear phishing is a targeted form of phishing where attackers customize their messages for a specific individual, group, or organization. Instead of sending mass emails, attackers conduct research on the victim (job role, company, personal interests) to craft highly convincing messages. Example: An attacker sends a fake email to a company's finance manager, pretending to be the CEO, requesting urgent funds transfer.

Phishing primarily exploits the application layer of communication, using protocols such as SMTP (email), HTTP/HTTPS (websites), and sometimes VoIP or messaging protocols. The mechanism involves: Crafting a message that looks authentic; Delivery through email, websites, or compromised accounts; Exploitation when the victim clicks a link, downloads a file, or shares credentials; Execution, where malware runs or credentials are stolen; Impact, which can lead to identity theft, data breaches, or financial loss.

Organizations can defend against phishing and spear phishing through a multi-layered security approach, including: Firewalls - block malicious IPs and filter unauthorized traffic; IDS/IPS - monitor network traffic, detect, and stop malicious activity; Security Patches - fix vulnerabilities exploited by attackers; Encryption - protect data both in transit and at rest; Email Security Gateways - detect and block phishing attempts; Awareness Training - educate users to recognize and avoid phishing attempts.

Phishing and spear phishing remain critical cybersecurity threats because they exploit the weakest link in any system: human behavior. Over 90% of successful cyberattacks begin with a phishing email. The best protection comes from a layered defense strategy that combines technical tools (firewalls, IDS/IPS, encryption, patches) with strong user awareness training.

Key Differences Between Phishing and Spear Phishing

Aspect	Phishing	Spear Phishing
Target	Mass audience (hundreds/thousands)	Specific individual or company
Message Style	Generic, often poorly written	Highly customized, convincing
Difficulty to Detect	Easier to spot	Much harder to detect
Attacker Effort	Low – one template sent widely	High – requires research
Risk Level	Moderate	Very High

Please use the link below to verify and check this website for authentication purposes.
Click on the hyperlink to open the site in your browser:

Phishing URL Checker

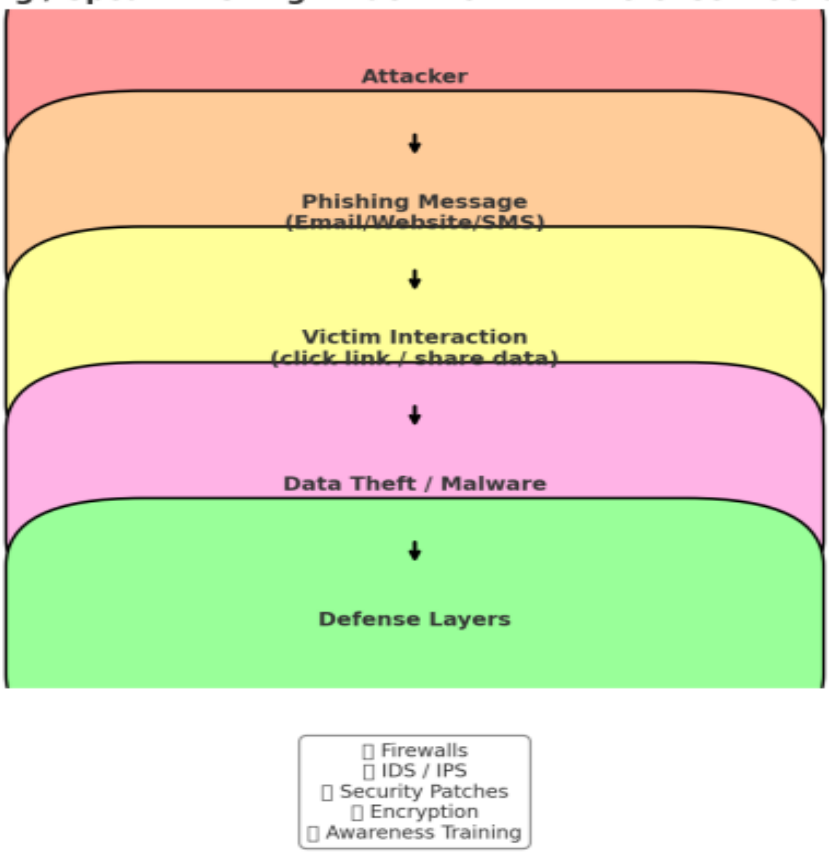
https://phishing-url-checker-l31y.vercel.app/

Trusted Domains

The following domains are considered trusted and safe for authentication. My website is designed to check against these trusted domains:

- **paypal.com**
- **google.com**
- **microsoft.com**
- **bankofamerica.com**
- **canvas.fau.edu**

Phishing / Spear Phishing Attack Flow with Defense Mechanisms



SQL Injection Attacks: What They Are and How They Work

SQL injection (SQLi) is a web security flaw that lets an attacker change the SQL queries an application sends to its database. When a website or API mixes user input directly into a query, rather than passing it safely as a parameter, malicious input can alter the query's logic that can expose sensitive records, change or delete data, or even run administrative commands on the database (OWASP).

At a high level, SQLi happens because code builds a string like "SELECT * FROM users WHERE username = '" + user + "' AND password = '" + pass + "';" and then sends it to the database. If an attacker enters a crafted value such as ' OR '1'='1, the WHERE clause becomes an always-true condition, bypassing checks. Other common flavors include UNION-based injection, which appends extra result sets from other tables; error-based injection, which forces the database to leak details through error messages; and blind/inference techniques, which use true/false questions or time delays to extract data when the app shows no errors (PortSwigger).

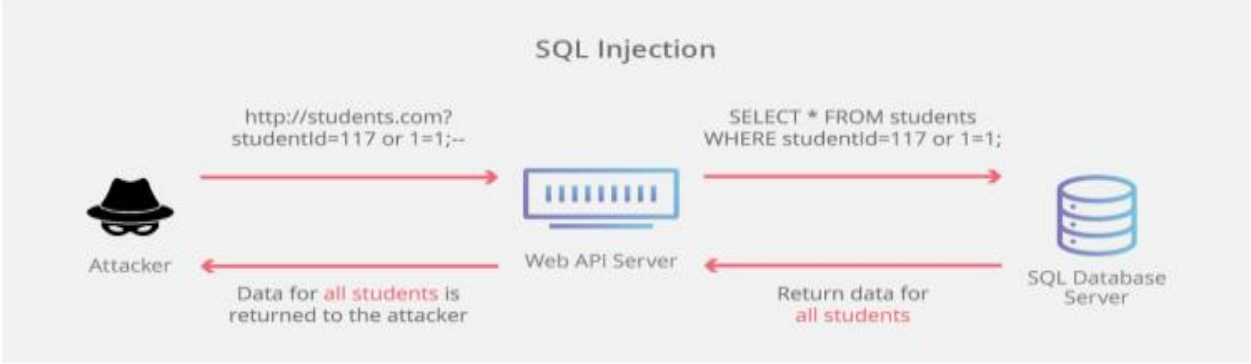
A well-known example is the 2015 breach of the UK telecom firm TalkTalk. Attackers exploited SQL injection on a legacy page to reach the customer database. Regulators later fined the company £400,000, citing outdated software and failure to secure known-vulnerable pages; the case is frequently used to show how one unpatched endpoint can expose an entire dataset (ICO).

SQLi succeeds for a few recurring reasons. First, developers sometimes trust input and concatenate it directly into queries. Second, database accounts used by the app often have broad privileges, so a single foothold opens far more data than necessary. Third, verbose error messages and weak monitoring make it easier for attackers to probe and harder for defenders to

spot abuse. Across industry taxonomies, SQL injection consistently ranks among the most dangerous weaknesses because it is both common and high-impact (OWASP).

Detection and prevention go together. Teams can uncover SQLi with code review that searches for raw string building, with automated dynamic tests that submit quotes and Boolean probes, and with monitoring that flags unusual query volume or timing behavior. The most effective mitigation is to use parameterized queries (prepared statements) or safe ORM bindings everywhere, which keep data separate from code. Complement this with server-side validation, minimal database privileges, prompt patching of frameworks and drivers, and non-verbose error handling. A web application firewall can add pattern-based blocking, but it should reinforce, not replace, secure coding (OWASP; PortSwigger).

Ultimately, SQLi persists because it is easy to introduce and hard to notice in complex codebases. Even mature stacks can reintroduce risk through unstructured query builders or unsafe string insertion. A practical baseline is simple: treat all input as hostile; parameterize every query; restrict database roles; review code that crosses trust boundaries; and test continuously in CI/CD with scanners and unit tests that verify parameters are used. Organizations that inventory legacy endpoints and retire or patch them quickly avoid “forgotten page” incidents like TalkTalk’s (ICO).



DNS Spoofing and Cache Poisoning: Undermining Internet Trust

The Domain Name System (DNS) is a critical component of the internet's infrastructure, functioning as a phonebook that translates human-readable domain names into numerical IP addresses. However, this system's foundational design lacked robust security, leaving it vulnerable to attacks such as DNS spoofing. DNS spoofing is a broad term for any attack that corrupts this resolution process to return an incorrect IP address. A specific and potent method to achieve this is DNS cache poisoning, where an attacker injects fraudulent DNS records into a resolver's temporary storage, or cache (Kessler, 2022). Once the cache is poisoned, the resolver will provide the false IP address to all users who query it for that domain, seamlessly redirecting them to a server controlled by the attacker.

This traffic redirection works by exploiting the stateless and trust-based nature of the DNS protocol. An attacker typically sends a flood of forged DNS responses to a target resolver, pretending to be from a legitimate authoritative name server. If the attacker successfully guesses critical query parameters like the transaction ID, the resolver will accept the fraudulent response and cache the malicious IP address. A landmark real-world example of this threat was the Kaminsky vulnerability discovered in 2008. Dan Kaminsky uncovered a flaw that allowed attackers to efficiently poison the cache not just for a single subdomain, but for an entire domain and all its services, such as web and email (Kaminsky, 2008). This critical flaw prompted a massive, secretive patching effort across the industry before its public disclosure.

To combat these threats, several defenses have been developed. DNSSEC (Domain Name System Security Extensions) is a suite of specifications that uses digital signatures to allow resolvers to verify the authenticity and integrity of DNS data (ICANN, 2013). Additionally, using secure, well-maintained public resolvers and encrypted DNS protocols like DNS-over-

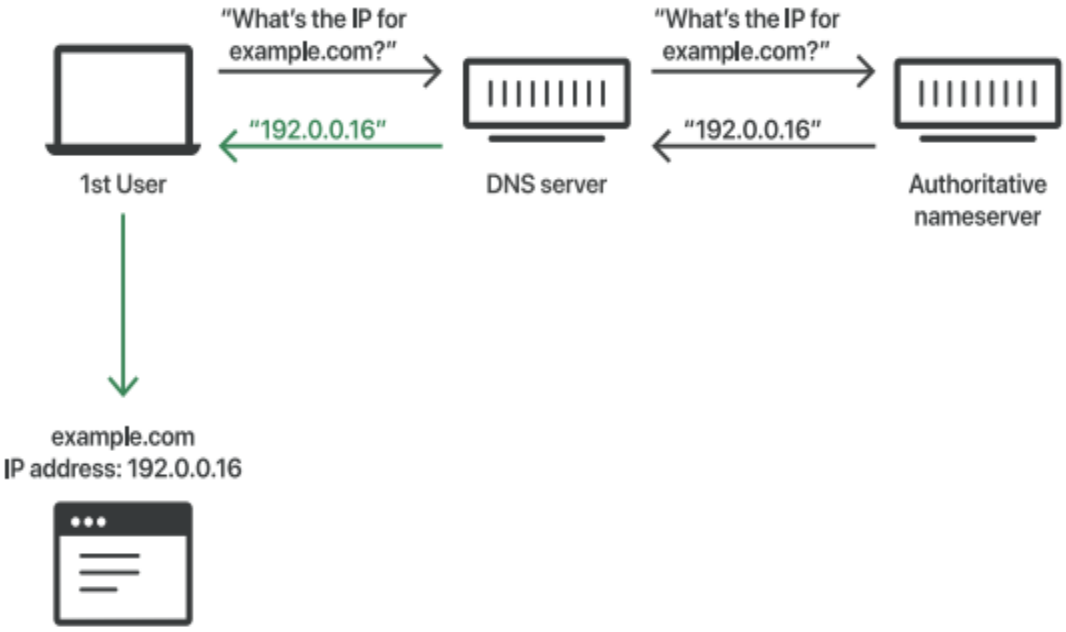
HTTPS (DoH) can prevent eavesdropping and on-path manipulation. However, a significant weakness is that DNSSEC is not yet universally adopted, leaving many domains and networks unprotected against sophisticated poisoning attacks.

This attack matters profoundly to communication networks because it strikes at the core of digital trust. By compromising the directory service that all network communication relies upon, attackers can intercept emails, redirect web traffic, and compromise any online service. This enables widespread phishing, espionage, and data theft on a massive scale, demonstrating that the security of the entire internet ecosystem is dependent on the integrity of this decades-old protocol.

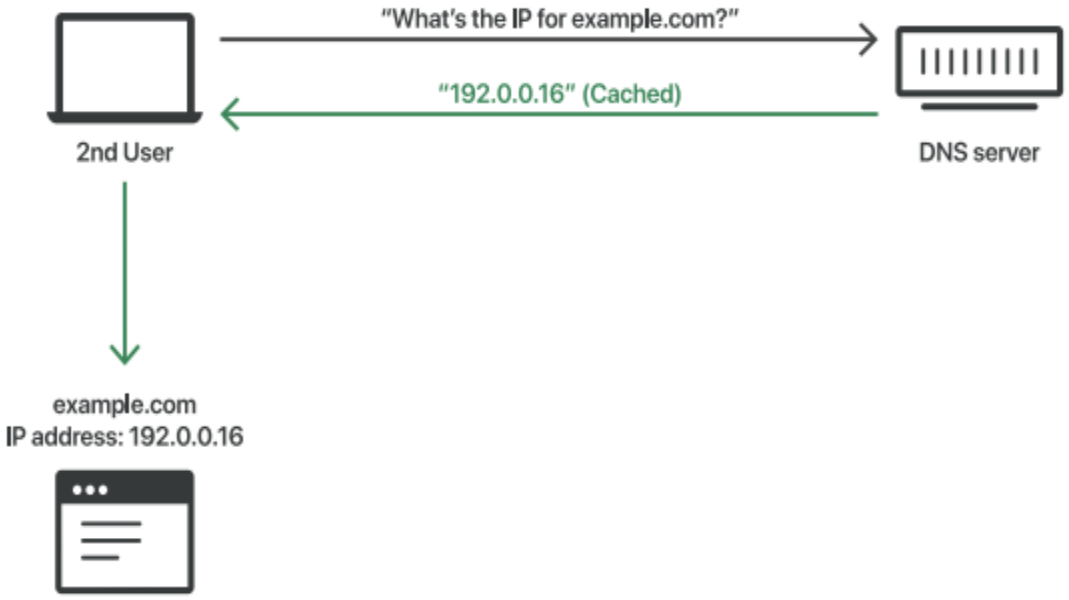
Visual Aids

How DNS Cache Poisoning Works (can be placed after first paragraph)

DNS Uncached Response:

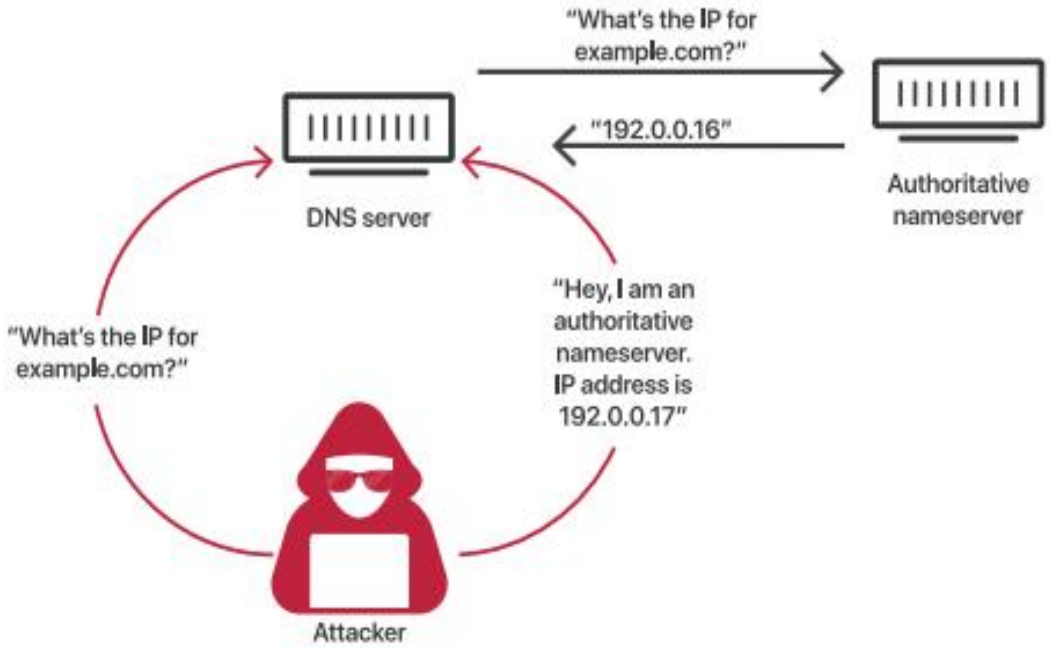


DNS Cached Response:

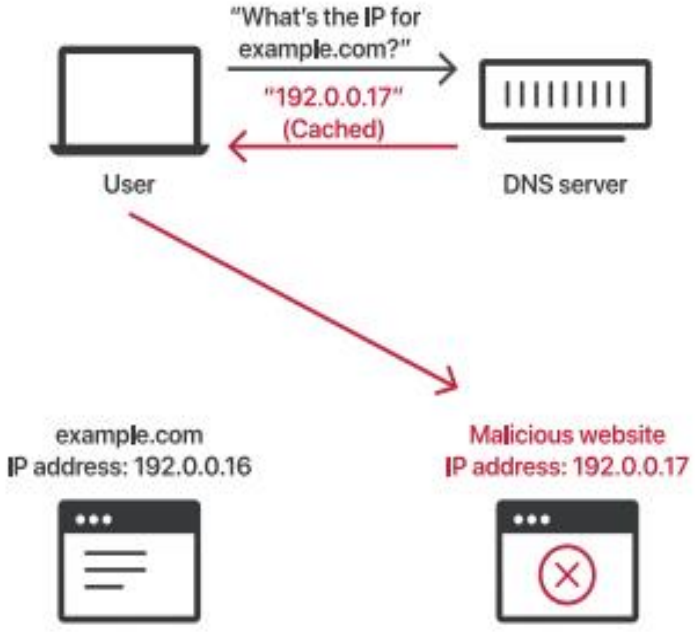


How DNSSEC Protects Against Spoofing: (goes with paragraph 3)

DNS Cache Poisoning Process:



Poisoned DNS Cache:



Man-in-the-Middle (MITM)

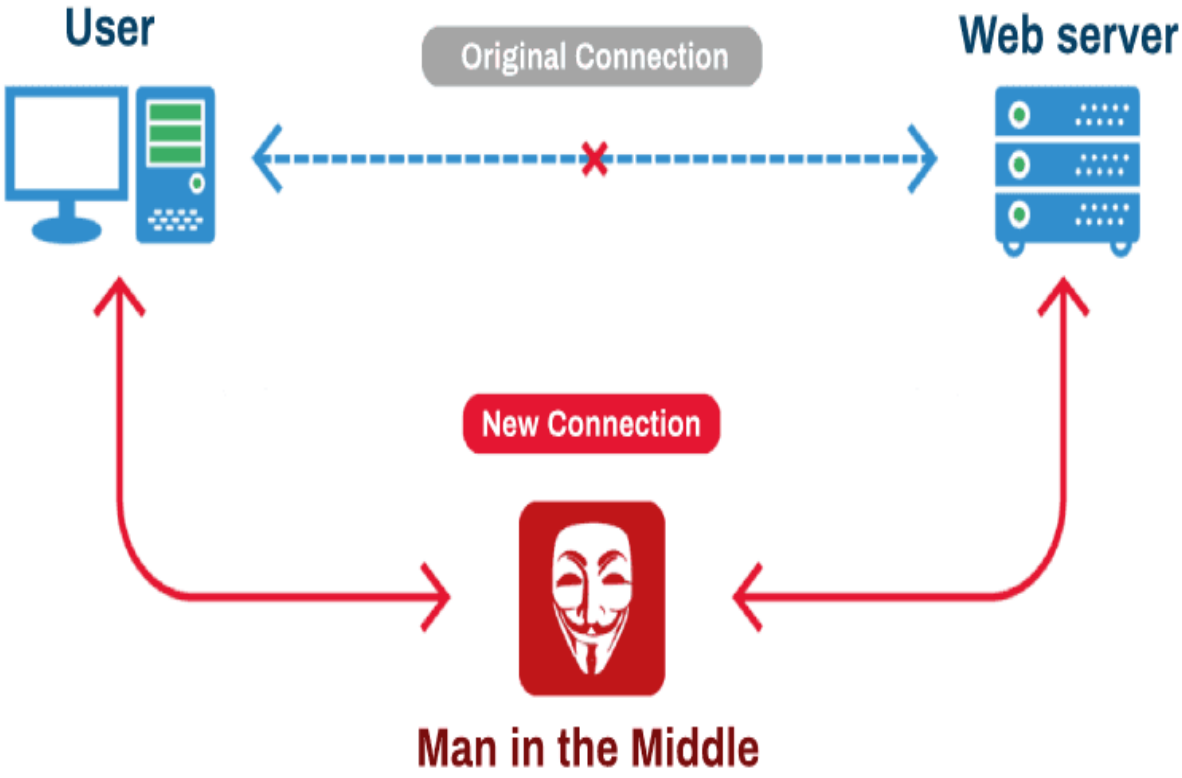
Man-in-the-Middle (MITM) attacks are another common method for attackers to act maliciously in communication networks. A MITM attack is defined as any attack in which an attacker intercepts and modifies communication data to impersonate a legitimate party (Das & Samdaria, 2014). One noteworthy type of MITM attack is the common Website Spoofing, also known as a Homograph Attack, in which an attacker tricks users into accessing an imposter site. Exacerbated using "Punycode", simplified display for complex Unicode characters, with which a URL may be spoofed despite identical appearance to its legitimate address (McCarthy, 2017). Additionally, within the broader Website Spoofing, a more specific attack vector called HTTPS Spoofing involves an attacker acquiring counterfeit SSL/TLS certificates to appear even more official, now sporting a padlock on the browser's address bar to "prove" authenticity (Twingate, 2025). An altogether different MITM strategy can be employed by a hacker to gain access to user data using a disguised wireless network, known as an Evil Twin. By setting up an often unsecured network with an identical name as a legitimate one, the attacker intends to trick users into connecting to their network and subsequently collect any unencrypted data and credentials the user transmits while connected (Kaspersky, 2025).

MITM attacks, due in part to their simple overarching structure, have a long history; within which the 2011 DigiNotar compromise is of considerable note. This hack exploited a vulnerability in the Dutch certificate authority, DigiNotar, allowing attackers to set up numerous MITM attacks using seemingly valid SSL certificates to impersonate major sites, notably including Google Gmail (Zetter, 2011). Another interesting MITM attack was conducted by the Allies' Royal Air Force during WWII: Aspidistra. This "Aspidistra" was a large radio transmitter, set up to engage with German communication networks and relay incorrect

information to German listeners. By mimicking German signals, the Allies were able to create widespread interference and mislead civilians and fighters, very closely mirroring more modern applications such as HTTPS Spoofing (Burden, 2008).

Despite being a rather robust hack, MITM attacks do have a variety of defenses: key-agreement protocols being chief among them. SSL/TLS is a common key-agreement protocol that incorporates a certificate authentication system which, when used with an HTTP communication, results in the ubiquitous web-security format known as HTTP over SSL/TLS (HTTPS). With sufficiently complex public/private key encryption, and proper certificate validation, HTTPS provides significant protection against MITM by hiding transmitted data and ensuring legitimate communication (Peterson & Davie, 2021). Other methods that an end-user may implement to mitigate some of the risk of MITM attacks include disabling Punycode display in their browsers (Twingate, 2025) and using Virtual Private Networks (VPNs) to encrypt their communications (Kaspersky, 2025).

Despite these numerous potential defenses, users may still enter an incorrect URL, fail to recognize a missing TLS certification on a given site, or connect to an unsecured or spoofed network. Any of these causes may be resisted by proper web-safety education, but no amount of care on users' end can prevent every manner of attack. Any MITM attack provides real and present danger to users of internet communication networks; they pose severe risks to privacy and data integrity and necessitate countless security protocols and methods across these networks to prevent irreparable damages.


User
Original Connection
Web server
New Connection
Man in the Middle

Conclusion

Cyberattacks against communication networks remain a constant, multi-layered risk to confidentiality, integrity, and availability. Ransomware, zero-day exploits, phishing and spear phishing, SQL injection, DNS spoofing, and man-in-the-middle (MITM) attacks target different points along the technology–people–process spectrum, but their effects often lead to disrupted services, compromised data, and eroded trust. Incidents such as ransomware campaigns against critical infrastructure and supply chain exploits show how one weak link can spread across interconnected systems. Because defense, public safety, and commerce depend on trustworthy communications, strengthening these networks is essential.

The best path forward is layered defense aligned to current best practices. Technical controls, zero-trust segmentation, authentication, encryption, secure coding, DNSSEC, certificate validation, and behavior-based detection must be paired with disciplined patch and configuration management. Training and phishing simulations reduce human error. Continuous monitoring, incident response, and backups build resilience through containment, recovery, and learning.

Resilience is a shared responsibility. Mission owners, service providers, and vendors should follow common standards, share indicators of compromise, and coordinate to detect attacks faster and limit their spread. By treating security as a continuous engineering process that is measured, audited, and improved, organizations enable communication networks to resist evolving threats rather than react to them. In conclusion, protecting the systems that carry our commands, commerce, and conversations requires consistent investment in both technology and culture. That dual commitment is the clearest route to maintaining reliable communications and national cybersecurity readiness. Ultimately, national security and overall data integrity rely on the resilience of our communication networks.

References

Begovic, A., Milosavljevic, M., & Djuric, M. (2023). *Detecting ransomware behavior through file-system and API monitoring.* arXiv. https://arxiv.org/abs/2306.12008

BlueVoyant. (2024). *Defense industrial base supply chain security report.* https://www.bluevoyant.com

Check Point Software Technologies. (2024). *Ransomware explained.* https://www.checkpoint.com

Cybersecurity and Infrastructure Security Agency [CISA]. (2021). *DarkSide ransomware: Impact on Colonial Pipeline.* https://www.cisa.gov/news-events/alerts/2021/05/10/darkside-ransomware-impact-colonial-pipeline

Cybersecurity and Infrastructure Security Agency [CISA]. (2024). *Stop ransomware guide.* https://www.cisa.gov/stopransomware

CrowdStrike. (2024). *Ransomware: Definition and prevention.* https://www.crowdstrike.com

Department of Homeland Security [DHS]. (2021). *Colonial Pipeline incident overview.* https://www.dhs.gov/news/2021/05/12/colonial-pipeline-incident-overview

Federal Bureau of Investigation [FBI]. (2024). *Common frauds and scams: Ransomware.* https://www.fbi.gov

National Institute of Standards and Technology [NIST]. (2023). *Ransomware basics for small businesses.* https://www.nist.gov/itl/smallbusinesscyber/guidance-topic/ransomware

IBM. (2023, June 2). *Zero-day.* IBM. https://www.ibm.com/think/topics/zero-day

Tubin, G. (2019, July). *Zero-day attacks, exploits, and vulnerabilities: A complete guide.* All-In-One Cybersecurity Platform – Cynet. https://www.cynet.com/zero-day-attacks/

Malwarebytes. (2023). *What is Stuxnet?* Malwarebytes. https://www.malwarebytes.com/stuxnet

Information Commissioner's Office. (2025, October 16). *TalkTalk cyber attack – How the ICO's investigation unfolded.* https://ico.org.uk/about-the-ico/media-centre/talktalk-cyber-attack-how-the-ico-investigation-unfolded/

OWASP Foundation. (2025, October 16). *SQL injection.* https://owasp.org/www-community/attacks/SQL_Injection

PortSwigger. (2025, October 16). *What is SQL injection? Tutorial & examples.* Web Security Academy. https://portswigger.net/web-security/sql-injection

Cloudflare. (2025, October 16). *SQL injection infographic.* https://www.cloudflare.com/img/learning/security/threats/sql-injection-attack/sql-injection-infographic.png

ICANN. (2013). *DNSSEC – What is it and why is it important?* https://www.icann.org/en/system/files/files/dnssec-qaa-09jan13-en.pdf

Kaminsky, D. (2008). *Black ops 2008: It's the end of the cache as we know it.* Black Hat USA.

Kessler, G. C. (2022). *An overview of DNS security.* GaryKessler.net. https://www.garykessler.net/library/dnssec.html

Cloudflare. (n.d.). *What is DNS cache poisoning?* Cloudflare Learning. https://www.cloudflare.com/learning/dns/dns-cache-poisoning/

Burden, M. R. (2008). *News, politics, sports, mail & latest headlines.* AOL. https://web.archive.org/web/20081009155628/http://members.aol.com/skywave48/aspidistra.htm

Das, M. L., & Samdaria, N. (2014). *On the security of SSL/TLS-enabled applications. Applied Computing and Informatics, 10*(1–2), 68–81. https://doi.org/10.1016/j.aci.2014.02.001

Kaspersky. (2025). *What is an evil twin attack? Evil twin Wi-Fi explained.* https://www.kaspersky.com/resource-center/preemptive-safety/evil-twin-attacks

McCarthy, K. (2017). *That apple.com link you clicked on? Yeah, it's actually Russian. The Register.* https://www.theregister.com/2017/04/18/homograph_attack_again

Peterson, L. L., & Davie, B. S. (2021). *Computer networks: A systems approach* (6th ed.). Morgan Kaufmann.

Twingate. (2025). *What is HTTPS spoofing? How it works & examples.* https://www.twingate.com/blog/glossary/https%20spoofing

Zetter, K. (2011). *DigiNotar files for bankruptcy in wake of devastating hack. Wired.* https://www.wired.com/2011/09/diginotar-bankruptcy