IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

A Study of Types of Hacking: Network, Operating System, and Application-Level Attacks

Priti Mane and Samruddhi Jogdand
Department of Information Technology
Government Polytechnic, Awasari (Khurd), Maharashtra, India

Abstract: *Hacking refers to identifying and exploiting weaknesses in computer systems, networks, operating systems, firmware, or applications. Depending on intent and authorization, hacking may be used for security testing or for malicious activity. This paper presents an overview of network hacking, operating system hacking, OS downgrade attacks, firmware-level exploits, application hacking, and advanced Web Application Firewall (WAF) bypass attacks. It explains common attack surfaces, potential impacts, and defensive practices. The discussion is based on published cybersecurity guidance and is intended for awareness and defensive learning rather than instructions for unauthorized access. The review indicates that layered security, timely updates, secure configuration, strong access control, monitoring, and independent application testing are important for reducing risk.*

Keywords: Network Security, Operating System Security, Downgrade Attack, Firmware Security, Application Security, WAF.

I. INTRODUCTION

Digital services depend on interconnected networks, operating systems, embedded firmware, and web applications. A weakness in any one layer can expose information, interrupt services, or allow unauthorized control. The term hacking is broad: authorized security professionals use controlled testing to discover weaknesses, while malicious actors may exploit similar weaknesses to steal data, disrupt operations, or maintain unauthorized access. Understanding the different layers helps organizations choose appropriate preventive and detective controls [1], [2].

This paper reviews three broad areas—network, operating system, and application hacking—with focused discussion of OS downgrade attacks, firmware-level exploits, and advanced WAF bypass risks. The scope is descriptive and defensive. No live systems were tested and no original attack experiments are claimed.

II. NETWORK HACKING

Network hacking involves attempting to access, observe, disrupt, or manipulate devices and data communicated through wired or wireless networks. Common risk areas include weak passwords, exposed services, insecure Wi-Fi settings, outdated network devices, poor segmentation, and unencrypted traffic. Threats may include unauthorized access, traffic interception, spoofing, denial of service, and lateral movement between systems.

The impact can include data exposure, service downtime, and compromise of connected devices. Defensive measures include changing default credentials, using strong authentication and modern encryption, closing unnecessary ports, segmenting sensitive systems, updating routers and switches, logging network activity, and monitoring unusual traffic. Network design should follow least privilege and should not assume that internal traffic is automatically trustworthy [1], [3].

III. OPERATING SYSTEM HACKING

Operating system (OS) hacking targets weaknesses in the software that manages hardware, memory, files, processes, users, and permissions. Vulnerabilities may arise from missing security updates, insecure configuration, excessive

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

privileges, unsafe drivers, or unsupported OS versions. Successful exploitation can result in privilege escalation, unauthorized access to files, persistence, or service disruption.

IV. APPLICATION HACKING

Application hacking targets weaknesses in software such as websites, mobile apps, APIs, and their supporting services. Common weaknesses include injection flaws, broken authentication, insecure access control, security misconfiguration, unsafe file handling, and exposed sensitive information. An attacker may attempt to manipulate application inputs, reuse stolen credentials, or access functions that should be restricted.

Secure development practices include validating input on the server, using parameterized database queries, applying authorization checks to every sensitive operation, protecting session tokens, storing secrets securely, and reviewing dependencies. Security testing should cover both automated checks and manual verification. The OWASP Web Security Testing Guide provides a structured approach for assessing application security and documenting findings [5].

V. ADVANCED WAF BYPASS ATTACK

A Web Application Firewall (WAF) filters HTTP/HTTPS requests in an attempt to detect and block malicious web traffic. A WAF is a useful defensive layer, but it cannot guarantee that an application is secure. A bypass occurs when a harmful request is not detected or blocked by the configured rules. This may happen because of inconsistent request parsing, incomplete rules, encoding or normalization differences, incorrect configuration, business-logic weaknesses, or a backend server that can be reached without passing through the WAF [6].

The term “advanced WAF bypass” describes attempts to evade filtering through complex or changing request patterns; it is not one single standardized attack. Defensive assessment should be performed only with authorization and in a controlled environment. Recommended controls include keeping WAF rules and managed rule sets updated, normalizing requests consistently, restricting direct access to origin servers, testing against representative attack cases, logging blocked and allowed requests, and fixing the underlying application flaw instead of relying on the WAF alone [5], [6].

VI. COMPARATIVE OVERVIEW

Network, operating system, firmware, and application attacks affect different layers, so the defensive controls should be selected according to the system and its risks. Table I summarizes the main risk and a corresponding defensive focus.

VII. CONCLUSION

Network, operating system, firmware, and application attacks affect different layers but can lead to related outcomes, including data loss, unauthorized control, and service interruption. Downgrade attacks exploit the continued use of weaker versions, while firmware attacks target a privileged layer that may be harder to inspect. Application attacks exploit weaknesses in software logic or implementation, and WAF bypasses demonstrate why perimeter filtering alone is insufficient. A defense-in-depth approach—combining secure configuration, timely patching, trusted firmware updates, least-privilege access, monitoring, and authorized security testing—offers stronger protection than any single control.

A. OS Downgrade Attack

An OS downgrade attack attempts to make a device use an older operating-system version, security protocol, or component that may contain known weaknesses. The term can also describe protocol downgrade attacks, where an attacker interferes with negotiation so that communicating parties use a less secure version. For example, a system that continues to support obsolete security protocols may expose data to risks that newer versions were designed to reduce [4].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

Risk reduction includes disabling obsolete protocols and unsupported versions, enforcing secure update policies, using signed updates and secure boot where supported, restricting rollback to approved recovery procedures, and monitoring unexpected version changes. Administrators should verify compatibility before disabling legacy support.

B. Firmware-Level Exploits

Firmware is low-level software that helps hardware components start and operate. Examples include system firmware, UEFI/BIOS, and firmware in network or peripheral devices. Firmware-level exploits may involve abusing vulnerable update mechanisms, unauthorized firmware modification, or flaws in privileged firmware services. Because firmware can operate beneath the OS, a compromise may be difficult for ordinary endpoint tools to detect and may persist through some software-level remediation.

NIST recommends protecting platform firmware against unauthorized changes, detecting changes, and supporting secure recovery [2]. Practical safeguards include verified vendor updates, cryptographic signatures, secure boot, restricted update permissions, asset inventories, and a tested recovery process. Firmware should only be obtained from trusted vendors and installed according to manufacturer guidance.

Attack area	Main risk	Key defensive focus
Network hacking	Unauthorized access or traffic interception	Segmentation, encryption, authentication, monitoring
OS hacking / downgrade	Privilege abuse or weaker versions	Patching, secure configuration, rollback controls
Firmware exploits	Low-level compromise and persistence	Signed updates, secure boot, verified recovery
Application hacking	Data exposure or unauthorized functions	Secure coding, access control, security testing
WAF bypass	Malicious requests reaching the application	Correct WAF configuration plus backend hardening

REFERENCES

[1] National Cyber Security Centre, "Network Security Fundamentals." Available: https://www.ncsc.gov.uk/guidance/network-security-fundamentals

[2] A. Regenscheid, "Platform Firmware Resiliency Guidelines," NIST Special Publication 800-193, May 2018. Available: https://doi.org/10.6028/NIST.SP.800-193

[3] National Security Agency, "NSA Details Network Infrastructure Best Practices," Mar. 1, 2022. Available: https://www.nsa.gov/Press-Room/News-Highlights/Article/Article/2949885/nsa-details-network-infrastructure-best-practices/

[4] U.S. Web Design System, "The HTTPS-Only Standard: Technical Guidelines." Available: https://https.cio.gov/technical-guidelines/

[5] OWASP Foundation, "Web Security Testing Guide." Available: https://owasp.org/projects/web-security-testing-guide

[6] OWASP Foundation, "OWASP Web Application Firewall Projects." Available: https://waf.owasp.org/