Benchmark – Lab 4: Your Second Hack Part 2

Ryan Coon

CYB-610

Professor Ian Standefer

April 23, 2025

**What is Burp Suite tool and how can it be used to help secure a website?**

Burp Suite is a Java-based suite of web application penetration testing tools, widely used by security professionals to assess and secure online applications. Employed by 90% of security experts, it authenticates incoming traffic, protecting websites from unsecured communication and potential vulnerabilities.

**What does CA stand for in the lab? What is it and why is it important?**

Certificate Authorities (CAs) are the organizations that issue SSL certificates—digital files cryptographically linking an entity to a public key. These certificates are the foundation of trust online, letting browsers verify that data from websites is legit. Burp Suite, a penetration testing tool, creates its own CA certificate when you first run it, storing it locally. This lets Burp intercept HTTPS traffic. Normally, your browser would freak out if Burp's self-signed certificate wasn't trusted, showing a security warning because it's not the real website's certificate. To make Burp work seamlessly with HTTPS sites, you add Burp's CA certificate to your browser's trusted root certificates. This way, Burp can intercept traffic without triggering security warnings, allowing you to inspect and test the site's security. This process protects against things like fake websites and man-in-the-middle attacks by encrypting data and verifying its integrity.

**What is a SQL injection attack?**

A SQL injection attack is a type of cyberattack that targets data-driven applications by injecting malicious SQL code into input fields, which can manipulate the application's database. This technique exploits vulnerabilities in an application's software, allowing attackers to execute arbitrary SQL commands. As a result, they can gain unauthorized access to sensitive data, modify or delete records, and even execute administrative operations on the database. SQL injection is one of the most prevalent web hacking techniques due to its simplicity and effectiveness. For instance, an attacker might input a specially crafted SQL statement into a login form, bypassing authentication and gaining access to the system.

**Provide 3 SQL injection string examples.**

' OR 1=1-- '

SELECT \* FROM users WHERE username = ‘admin’ AND password = ‘ ’

DROP TABLE users--

**Provide a mitigation technique and a technology that can be used to offset this vulnerability.**

To effectively mitigate SQL injection attacks, one of the most recommended techniques is the use of prepared statements with parameterized queries. This approach ensures that SQL code and data are separated, preventing attackers from injecting malicious SQL code into input fields. By using prepared statements, the SQL query is defined with placeholders for parameters, which are then bound to actual values at execution time. This method not only enhances security but also improves performance by allowing the database to cache the execution plan.

In addition to prepared statements, employing Web Application Firewalls (WAFs) can significantly bolster defenses against SQL injection attacks. A WAF acts as a filter between the web application and the internet, monitoring and analyzing incoming traffic for malicious patterns. It can detect and block SQL injection attempts in real-time, providing an additional layer of security. Many modern WAFs come equipped with built-in rules specifically designed to identify and mitigate SQL injection threats.

**What mitigation techniques would you suggest if upgrading or replacing a vulnerable OS was not an option?**

If upgrading or replacing a vulnerable operating system (OS) is not feasible, there are several effective mitigation techniques that can be employed to enhance security and reduce risks. Here are some key strategies:

Network Segmentation - Isolate the vulnerable OS from critical systems and sensitive data by implementing network segmentation. This limits the potential impact of an attack, as it restricts lateral movement within the network. By creating separate network zones, you can control access and monitor traffic more effectively.

Implementing Firewalls and Intrusion Detection Systems (IDS) - Utilize firewalls to filter incoming and outgoing traffic based on predetermined security rules. Additionally, deploying IDS can help detect and respond to suspicious activities in real-time, providing an extra layer of defense against potential exploits targeting the vulnerable OS.

Regular Patching and Updates - While the OS itself may not be upgradable, ensure that all applications and services running on it are regularly updated and patched. This includes third-party software, which can often be a vector for attacks. Keeping these components up to date can mitigate known vulnerabilities.

Application Whitelisting - Restrict the execution of unauthorized applications by implementing application whitelisting. This approach allows only approved software to run on the system, significantly reducing the risk of malware infections and unauthorized access.

Enhanced Monitoring and Logging - Increase the level of monitoring and logging on the vulnerable OS. By closely monitoring system logs and user activities, you can quickly identify and respond to any unusual behavior that may indicate a security breach.

User Education and Awareness - Conduct training sessions for users to raise awareness about security best practices, such as recognizing phishing attempts and avoiding risky behaviors that could compromise the system. Educated users are less likely to inadvertently expose the system to threats.

Backup and Recovery Plans - Establish robust backup and recovery procedures to ensure that data can be restored in the event of a security incident. Regularly test these backups to confirm their integrity and effectiveness.

**Describe a secure network architecture that will protect your organization's internal systems from a system with a vulnerable OS.**

A secure network architecture to protect internal systems from a vulnerable OS employs a layered approach. This includes a strong perimeter with firewalls and intrusion detection/prevention systems, network segmentation to isolate vulnerable systems, strict access controls (least privilege, RBAC), endpoint protection (antivirus, EDR), data encryption, and continuous monitoring.

**How would you incorporate a defense in depth methodology and protect this system from outside interference?**

Implementing a defense-in-depth strategy is crucial for shielding a system from external interference. This approach involves layering multiple security controls to create a robust defense, recognizing that no single measure is foolproof. The perimeter of the system should be fortified with firewalls, intrusion detection and prevention systems (IDPS), and robust access controls to filter and scrutinize all incoming and outgoing network traffic, acting as the first line of defense. Network segmentation is then employed to divide the system into isolated zones, restricting the lateral movement of any potential attackers who manage to bypass the perimeter defenses. Within each segment, strict access control measures, based on the principle of least privilege and role-based access control, limit user and system access to only the necessary resources, further minimizing the impact of a compromise. Endpoint security, including antivirus software, host-based firewalls, and endpoint detection and response (EDR) solutions, is deployed on all devices to detect and mitigate threats at the device level. Data encryption, both at rest and in transit, protects sensitive information from unauthorized access, even if a system is breached. Regular vulnerability assessments and penetration testing are conducted to identify weaknesses and proactively address them. Security information and event management (SIEM) systems aggregate and analyze security logs from various sources, providing real-time visibility into potential threats and enabling rapid incident response. Furthermore, a well-trained workforce, educated on security best practices and threat awareness, serves as a critical human firewall, preventing social engineering and other attacks.

References:

Cisco. (2022, February). *Cisco Security Reference Architecture*. Cisco. https://www.cisco.com/c/en/us/products/security/cisco-security-reference-architecture.html

Firch, J. (2024, February 21). *Common Types Of Network Security Vulnerabilities In 2022*. PurpleSec; PurpleSec. https://purplesec.us/learn/common-network-vulnerabilities/

Fourie, M. (2022, July 18). *Top 5 Strategies for Vulnerability Mitigation*. Centraleyes. https://www.centraleyes.com/top-5-strategies-for-vulnerability-mitigation/

Kime, C. (2024, April 26). *Network Security Architecture: Best Practices & Tools*. ESecurity Planet. https://www.esecurityplanet.com/networks/network-security-architecture/

kingthorin, & zbraiterman. (2024). *SQL Injection*. OWASP. https://owasp.org/www-community/attacks/SQL\_Injection

National Security Agency. (2018). *NSA’S Top Ten Cybersecurity Mitigation Strategies*. https://www.nsa.gov/portals/75/documents/what-we-do/cybersecurity/professional-resources/csi-nsas-top10-cybersecurity-mitigation-strategies.pdf

OWASP. (2025). *SQL Injection Prevention · OWASP Cheat Sheet Series*. Owasp.org; Owasp. https://cheatsheetseries.owasp.org/cheatsheets/SQL\_Injection\_Prevention\_Cheat\_Sheet.html