Hands-On Application Security Testing: From Automation to Exploitation

This slide explores the full spectrum of Dynamic Application Security Testing (DAST), blending automated WebInspect scanning, manual BurpSuite testing, and custom hands-on labs. Attendees will learn to leverage a diverse toolkit to identify and exploit real-world web vulnerabilities, from discovering weaknesses to crafting effective remediations.

Evolving Application Vulnerability Scanning

AUTOMATED VULNERABILITY SCANNING WITH WEBINSPECT

I used WebInspect to automate the scanning of web applications for a wide range of vulnerabilities, including cross-site scripting (XSS), SQL injection, and unpatched software. By integrating automated scanning into our workflows, I was able to quickly identify and prioritize security issues across the application portfolio.

MANUAL PENETRATION TESTING WITH BURPSUITE

In addition to automation, I performed manual penetration testing with BurpSuite. I developed and applied custom testing techniques to uncover complex vulnerabilities, conduct deeper analysis, and validate the effectiveness of remediation strategies.

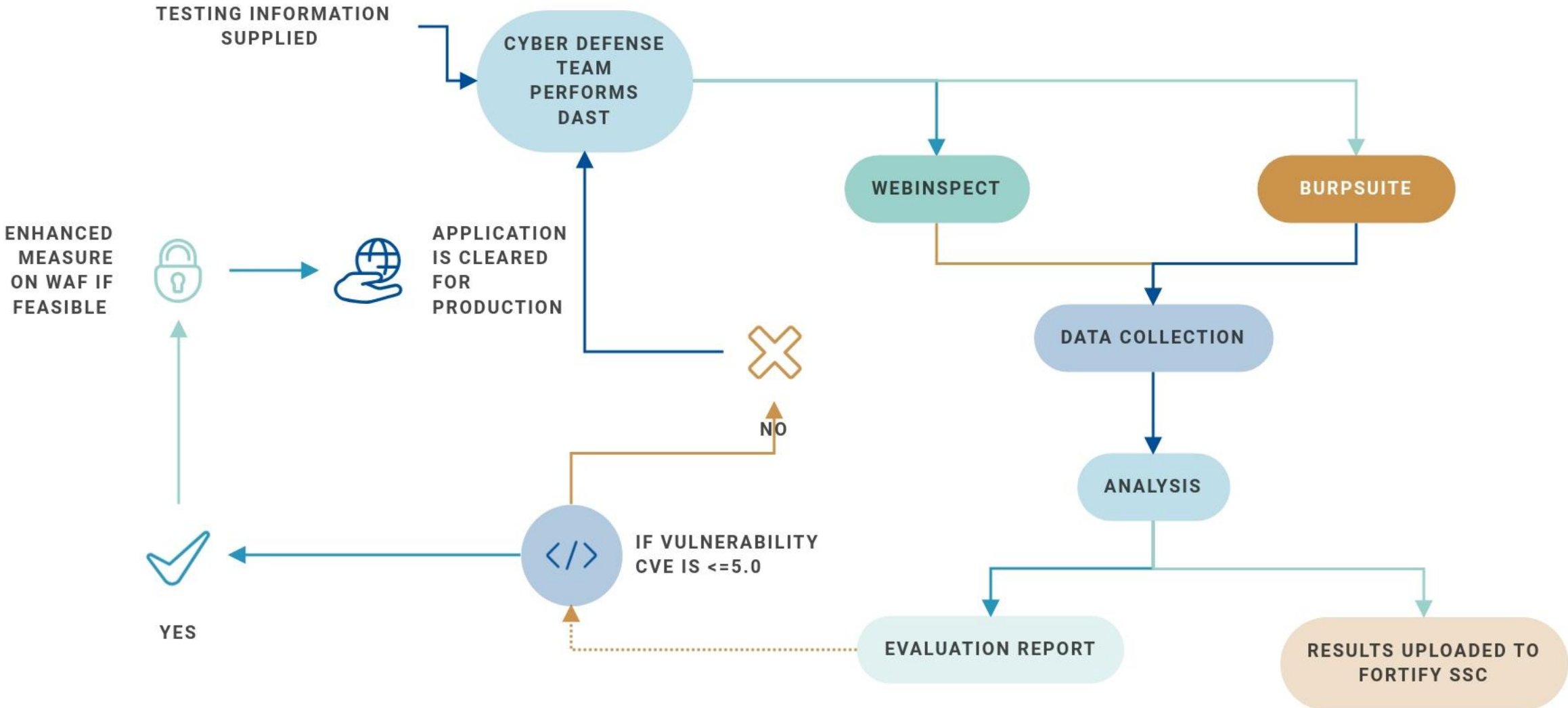
COMPREHENSIVE REMEDIATION ADVISORIES

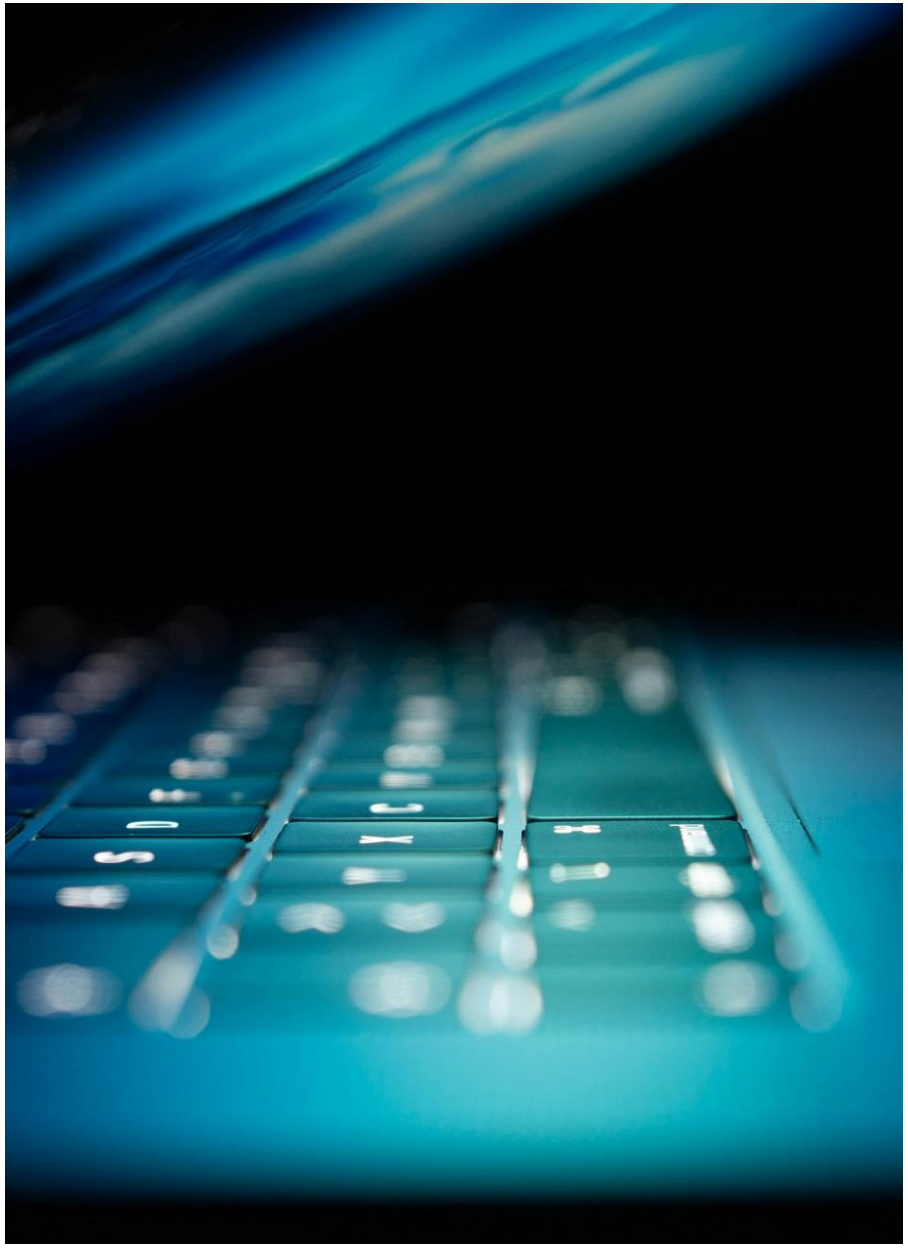
I also created remediation guidance for developers, providing detailed explanations of the root causes of vulnerabilities and recommending specific code changes or configuration updates.

COMPENSATING CONTROLS WITH WEB APPLICATION FIREWALL (WAF)

As a final layer of defense, I configured compensating controls within the Web Application Firewall (WAF). This helped detect and mitigate known vulnerabilities and attack patterns, ensuring applications remained protected while long-term fixes were being implemented.

DAST Scan Methodology





Application Vulnerability Labs: Dynamic Application Security Testing

Application Vulnerability Labs provides hands-on labs for exploring real-world web vulnerabilities. Dive into automated scanning, manual testing, and effective remediation strategies to enhance your cybersecurity skills.

Weak TLS Cipher Acceptance (CipherCheckTestLab)

- **WEAK TLS CIPHER ACCEPTANCE**

  Vulnerability: Weak SSL/TLS ciphers accepted by server – allows downgrade of encryption. Server misconfiguration may permit using obsolete cipher suites (e.g., 40-bit or RC4), risking exposure of sensitive data.

- **POC TOOL: CUSTOM BASH SCRIPT USING CURL --CIPHERS OPTION**

  Tests various cipher suites against the target. Automates checks across many ciphers (from strong to very weak) and logs which ones are accepted.

- **INJECTION FEATURES: SUPPORTS COOKIE AND URL INJECTION**

  Tester can specify a session cookie to simulate an authenticated user, and target specific URLs (endpoints) on the site. Ensures even pages requiring login can be tested for weak cipher enforcement.

- **OUTCOME: IF THE SERVER NEGOTIATES ANY DEPRECATED CIPHER**

  The script flags it. This confirms the server can be tricked into using a cipher that should be disallowed, potentially enabling attackers to decrypt or manipulate supposedly secure sessions.

- **REMEDIATION: DISABLE LEGACY CIPHERS ON THE SERVER SIDE**

  Enforce minimum TLS version and strong cipher suites (follow industry recommendations to prevent downgrade attacks).

Clickjacking Framing Attack (clickjacking-lab)

- **VULNERABILITY: CLICKJACKING (UI REDRESSING)**

  The attack page frames the target website and tricks the user into clicking invisible or disguised elements. The user believes they are interacting with the visible page, but their clicks are actually performing actions on the hidden target frame.

- **LAB SETUP: MALICIOUS HTML PAGE WITH**

  The iframe is styled (with CSS opacity or stacking) so that a critical action button on the target is positioned under a seemingly benign clickable element on the top page.

- **COOKIE INJECTION: PROGRAMMATICALLY SET SESSION COOKIE**

  The attacker's page can set or instruct the victim to set a session cookie for the target site, so that the framed action (e.g., "Delete Account") will execute as if the user was logged in.

- **DEMONSTRATED IMPACT: UNAUTHORIZED ACTIONS ON USER ACCOUNT**

  Without defenses like X-Frame-Options or frame-busting scripts, an attacker can force a user to perform unauthorized actions on their account, such as clicking a "Transfer Funds" button on the hidden bank site frame.

- **MITIGATIONS: DEVELOPERS SHOULD USE X-FRAME-OPTIONS, CSP**

  Developers should send the X-Frame-Options: DENY or SAMEORIGIN header, use the newer Content-Security-Policy: frame-ancestors directive, and ensure critical UI actions have additional confirmation.

HTTP Request Smuggling (Conflicting CL/TE) – WebInspect Lab

- **VULNERABILITY: HTTP REQUEST SMUGGLING**

  Exploiting inconsistent parsing of request boundaries by front-end vs. back-end servers. Including both Content-Length and Transfer-Encoding: chunked headers creates ambiguity in how the request is processed.

- **TECHNIQUE: CRAFTED HTTP MESSAGE WITH DECEPTIVE BODY**

  The attack uses a crafted HTTP message with a deceptive body that actually contains a second, smuggled HTTP request.

- **FRONT-END VS. BACK-END PROCESSING DISCREPANCY**

  The front-end (e.g., proxy, load balancer) uses one header to determine request length while the back-end uses the other, causing them to see different requests.

- **IMPACT: BYPASSING SECURITY CONTROLS**

  The successful exploit allows the attacker to inject a hidden request that the back-end will process as a separate HTTP request, bypassing security checks.

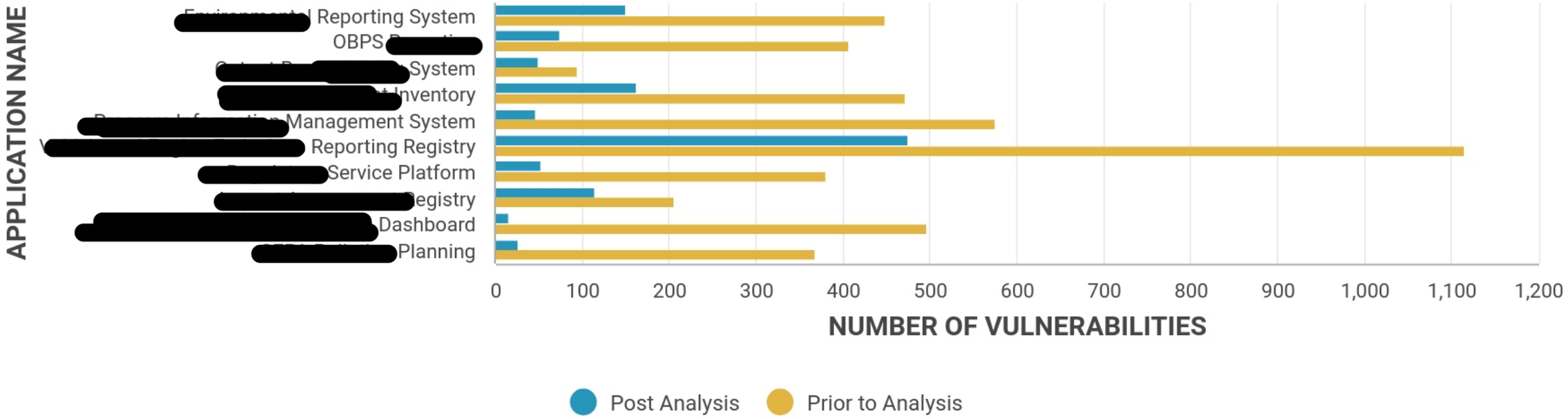
- **POTENTIAL IMPACTS**

  This can let the attacker steal data or perform actions in another user's session, effectively breaking the normal request isolation on the server.

- **REMEDIATION**

  Reject or properly handle ambiguous requests, update proxies/servers to the latest versions with known fixes, and ensure the front-end removes one of the headers when both are present.

Completed Projects



THE NUMBER OF VULNERABILITIES BEFORE AND AFTER THE ANALYSIS HAS BEEN DECREASING, WHICH POINTS TO ADVANCED DETECTION MEASURES AND A COMPREHENSIVE REPOSITORY OF KNOWN VULNERABILITIES

Key Takeaways

☆ THE ROLE OF AUTOMATED SCANNING (WEBINSPECT) FOR BROAD VULNERABILITY COVERAGE

Automated WebInspect scans provide speed and broad coverage to rapidly identify and prioritize a wide range of vulnerabilities across the application.

☆ HOW MANUAL PENETRATION TESTING (BURPSUITE) ADDS DEPTH BY UNCOVERING COMPLEX VULNERABILITIES AND VALIDATING FIXES

Manual BurpSuite testing complements automation by uncovering complex, real-world vulnerabilities through the application of custom techniques, deeper analysis, and validation of remediation strategies.

☆ REAL-WORLD ATTACK SCENARIOS THROUGH HANDS-ON LABS: CIPHER-CHECK, CLICKJACKING, AND HTTP REQUEST SMUGGLING

The hands-on labs demonstrate critical risks, such as weak TLS cipher acceptance, clickjacking, and HTTP request smuggling, allowing attendees to learn and practice exploiting these vulnerabilities.

☆ THE IMPORTANCE OF REMEDIATION STRATEGIES AND COMPENSATING WAF CONTROLS TO PROTECT APPLICATIONS DURING PATCH CYCLES

Effective remediation requires a layered approach, including WAF compensating controls and secure configurations, to ensure stronger protection while long-term fixes are being applied.