Incident Response Playbook: Web Shell Upload & Execution

Team AnubisX

Version 1.0
September 17, 2025

Document Control

Attribute	Value
Version	1.0
Status	Final
Owner	AnubisX Security Team
Review Cycle	Every Quarter

Contents

1 Introduction

1.1 Purpose

The purpose of this playbook is to provide a structured incident response plan for handling web shell upload and execution incidents, with the objective of identifying unauthorized web-based access, removing malicious web shells, and restoring web application integrity.

1.2 Scope

This playbook applies to web servers, web applications, associated backend systems, and administrative accounts. It covers all stages of incident response, from preparation to post-incident lessons learned.

2 Overview of a Web Shell Attack

A web shell is a script uploaded to a vulnerable web server that provides remote command execution and file access to an attacker. Web shells are commonly deployed through exploited file upload functionality, vulnerable web applications, or compromised admin panels. They enable persistent remote access, data theft, and further network intrusion if not detected and removed promptly.

3 Incident Response Phases

This playbook follows the NIST Incident Response lifecycle framework.

3.1 Phase 1: Preparation

Goal: To ensure the team is equipped and ready to respond to a web shell incident before it occurs.

- **Roles and Responsibilities:** Define roles (Incident Commander, Lead Analyst, Forensic Analyst, Web/App Security Lead, Communications Lead).
- **Tools & Resources:** Ensure availability of WAF, web server logs, EDR, SIEM, file integrity monitoring, and forensic tools.
- **Training:** Conduct web-app vulnerability exercises and simulated web shell detection drills.
- **Contact Lists:** Maintain updated contact lists for hosting providers, CMS vendors, and external IR partners.
- **Threat Intelligence:** Monitor indicators of web shell campaigns and known web shell signatures.

3.2 Phase 2: Identification & Analysis

Goal: To confirm web shell presence and determine its scope and impact.

1. **Initial Triage:** Collect web server logs, WAF alerts, and EDR telemetry. Open an official incident ticket and activate secure communication channels.
2. **Initial Analysis and IOC Evaluation:** Analyze logs and alerts to identify Indicators of Compromise (IOCs). Common IOCs include:

- **Web Logs:** Unusual POST requests to upload endpoints, requests with suspicious user-agents, access to uncommon script files (e.g., .php, .aspx) in uploads folders.
- **File System:** Presence of unknown script files, modified timestamps, unexpected permissions changes.
- **Network:** Outbound connections from web servers to suspicious IPs, use of uncommon ports.
- **Account:** Compromised admin credentials or unauthorized admin panel access.

3. **Severity Level Assessment:** Classify the incident to ensure appropriate allocation of resources. Severity is determined based on the operational impact, the criticality of the affected systems and data, and the scope of the compromise.

Level	Description	Example	MTTD	MTTR
Low	Single web shell instance detected in a non-production or low-risk application.	A test server's upload directory contains a single web shell file; no evidence of execution or lateral movement.	6–12 hours	24–48 hours
Medium	Web shell detected with limited successful execution on one production application.	A web shell is found in a public-facing app and logs show limited command execution but no data exfiltration.	12–24 hours	2–4 days
High	Active web shells across multiple web servers with evidence of data access or lateral movement.	Multiple web servers show web shell activity with file staging and outbound connections to attacker infrastructure.	24–48 hours	4–7 days
Critical	Widespread web shell deployment with confirmed data theft, admin takeover, or further network compromise.	Web shells installed on critical web servers and admin panels; attacker created backdoors and exported sensitive databases.	48 hours	7–14 days

Table 1: Incident Severity Matrix

4. **Alert Validation (TP vs. FP):** Correlate suspicious web activity with other logs and web-shell-specific threat intelligence.
   - **If True Positive (TP):** The activity is confirmed as a web shell. **Action:** Immediately proceed to the **Containment** phase, escalate to the Incident Commander, and activate the web shell playbook.
   - **If False Positive (FP):** The activity is confirmed benign. **Action:** Document findings, close the alert, and recommend tuning WAF/IDS rules.
5. **Incident Declaration:** If confirmed, formally declare a web shell incident and escalate to leadership, web/app teams, and hosting providers.

3.3 Phase 3: Containment

Goal: To limit the web shell's impact and prevent further damage.

- **Short-Term Containment (Immediate Actions):**
  - Immediately block upload endpoints or take the application out of service.
  - Isolate affected web servers from the network.
  - Revoke service account credentials and disable compromised accounts.
  - Block outbound connections to attacker IPs and domains.
- **Evidence Preservation:** Acquire forensic artifacts (web logs, uploads folder, web server memory) **before** remediation.
- **Long-Term Containment Strategy:** Segment web server networks and apply stricter firewall rules.

3.4 Phase 4: Eradication

Goal: To remove web shell components and prevent reinfection.

- **Root Cause Analysis:** Identify the entry vector (e.g., file upload vulnerability, CMS plugin exploit).
- **Malware Removal:** Reimage compromised web servers from known-good baselines and remove web shell files.
- **Persistence Removal:** Eliminate any backdoors, scheduled tasks, or additional malicious files left by the attacker.
- **Security Hardening:** Patch the underlying vulnerability, rotate credentials, and harden web server configurations.

3.5 Phase 5: Recovery

Goal: To safely restore web applications and business operations.

- **System Restoration:** Restore applications from clean backups or rebuild securely.
- **Enhanced Monitoring:** Increase monitoring of web logs and file integrity post-restoration.
- **Validation:** Ensure restored applications are secure and malware-free before reconnecting to the production network.
- **Business Continuity:** Coordinate with leadership to prioritize the restoration of critical web services.

3.6 Phase 6: Post-Incident Activities (Lessons Learned)

Goal: To strengthen resilience and prevent recurrence.

- **Post-Incident Meeting:** Conduct a blameless post-mortem with web, security, and operations teams.
- **Final Incident Report:** Create a detailed report covering the root cause, timeline, response actions, and impact.
- **Action Plan:** Create a tracked action plan to implement security improvements (e.g., secure code reviews, file upload controls, WAF tuning).

4 MITRE ATT&CK Framework Mapping

Web Shell Upload & Execution ATT&CK Mapping

- **Tactic: Initial Access**
  - *T1190 – Exploit Public-Facing Application*
  - *T1566 – Phishing*
  - *T1189 – Drive-by Compromise*
- **Tactic: Execution**
  - *T1059 – Command and Scripting Interpreter*
  - *T1204 – User Execution*
- **Tactic: Persistence**
  - *T1505.003 – Server Software Component: Web Shell*
  - *T1053 – Scheduled Task/Job*
- **Tactic: Privilege Escalation**
  - *T1068 – Exploitation for Privilege Escalation*
  - *T1078 – Valid Accounts*
- **Tactic: Defense Evasion**
  - *T1027 – Obfuscated Files or Information*
  - *T1562 – Impair Defenses*
- **Tactic: Credential Access**
  - *T1003 – OS Credential Dumping*
- **Tactic: Discovery**
  - *T1083 – File and Directory Discovery*
  - *T1018 – Remote System Discovery*
- **Tactic: Lateral Movement**
  - *T1021 – Remote Services*
  - *T1570 – Lateral Tool Transfer*
- **Tactic: Collection**
  - *T1530 – Data from Cloud Storage Object*
- **Tactic: Exfiltration**
  - *T1041 – Exfiltration Over C2 Channel*
- **Tactic: Impact**
  - *T1485 – Data Destruction*
  - *T1486 – Data Encrypted for Impact*