Cybersecurity Incident Report - Apex Financials (INC250422)

Group 1

Guided by – Dr.Premila Melvin, John Griffin

Group Members

01 Leela Pavan

02 Phanindhar Reddy

03 Junaid

04 Shalem Raju

05 Sriram

06 Srivarsha

Table of Contents

Executive Summary

01

- On April 19, 2025, Apex Financials' public-facing web server (192.168.5.200) was compromised.
- The attacker gained remote access via a combination of brute-force login attempts, SQL Injection, and deployment of a PHP web shell (c99shell.php).
- Over 50GB of sensitive data was exfiltrated via outbound HTTP traffic to attacker IP 167.172.3.114.

- **Targeted Asset:** Apache-based web server hosted in the DMZ.

- **Initial Attack Vector:**
  Brute-force attempts on /login.php
  SQL Injection used to bypass authentication
  Remote Code Execution via vulnerable PHP script eval-stdin.php

- **Threat Actor TTPs**:
  Reconnaissance, command injection, reverse shell
  Use of common web shell (c99shell.php)
  Lateral movement attempts detected in logs

Case Scope

02

- The investigation aimed to reconstruct the complete intrusion timeline, validate attacker behavior, and assess the overall impact on data and systems.

- **Log Sources Analyzed:**

  Web Server Access Logs (access.log)

  Firewall Logs (FW_logs.txt)

  Intrusion Detection Alerts (IDS_logs.txt)

- **Splunk was used for:**

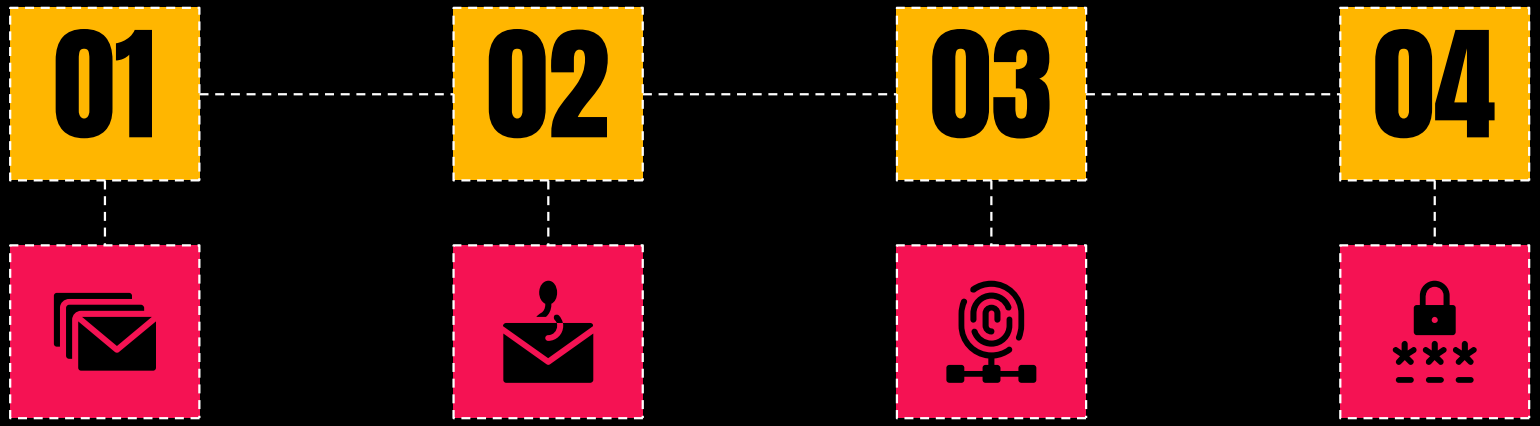
  Querying and correlating logs from multiple sources

  Reconstructing the second-by-second attack timeline

  Identifying Indicators of Compromise (IoCs)

  Confirming the data exfiltration path

Attack Timeline



Reconnaissance

Enumeration of login page

Exploitation

SQL injection, RCE via PHP shell

Privilege Escalation

Commands executed remotely

Data Exfiltration

50GB transferred via HTTP sessions

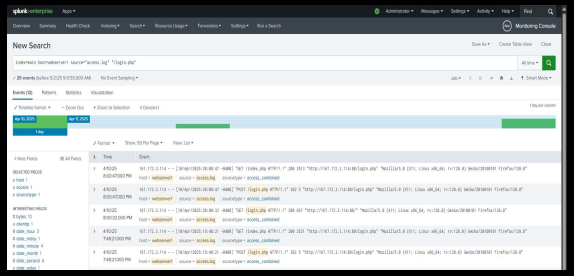
Root Cause Identification & Log Analysis 03

The attacker exploited weak server-side input validation and poor PHP configuration to gain access.

Initial Entry Point:

Brute-force login attempts on /login.php

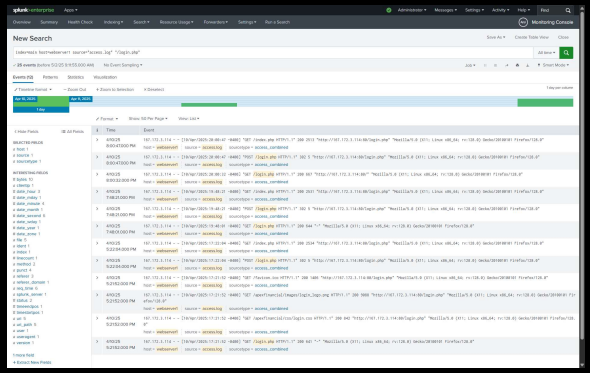
Successful SQL Injection used to bypass authentication



Post-Exploitation Activity:

Uploaded and executed PHP web shells (eval-stdin.php, c99shell.php)

Gained Remote Code Execution and listed server files (ls, whoami commands)



Key Findings from Logs:

access.log showed command executions via cmd= parameters

IDS_logs.txt flagged critical web shell and RCE alerts

FW_logs.txt confirmed large outbound traffic to 167.172.3.114

Scope and Impact of the Incident

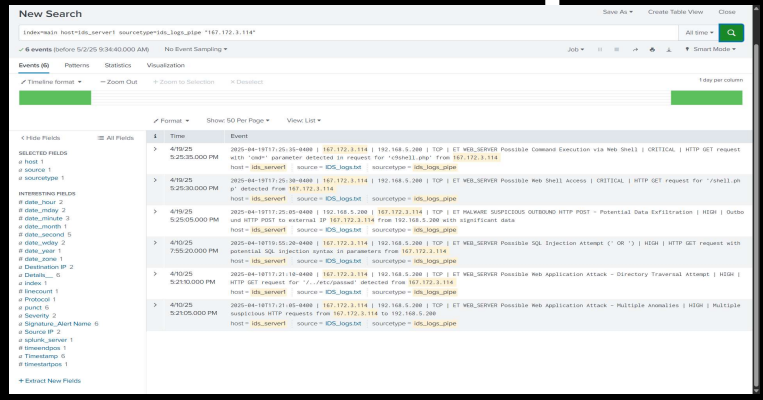
Impacted System: Public-facing web server – 192.168.5.200

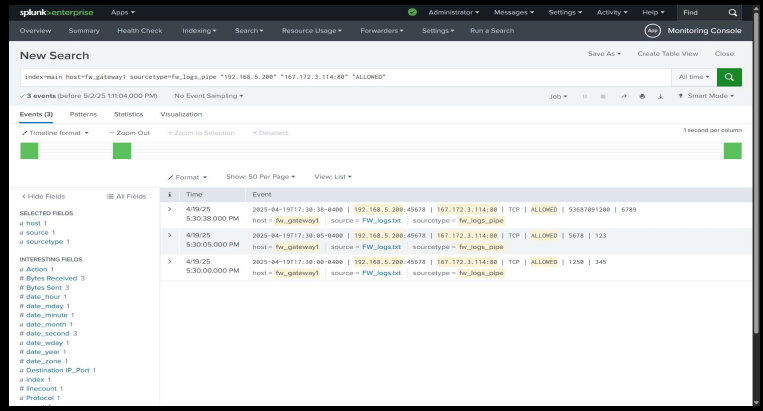
Duration of Attack Activity: From April 10 to April 19, 2025, based on log timestamps

Compromise Details:
Gained shell access via vulnerable PHP scripts
Executed system commands remotely
Accessed sensitive files, including db_config.php and backup.tar.gz

Data Exfiltration:
Over 50GB of data sent to attacker IP 167.172.3.114
Confirmed via outbound HTTP traffic in firewall logs

Potentially Exposed Information:
System backups
Database credentials
CSV/XLS files likely containing financial or user data





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

+

Containment Strategy

Immediate actions taken to isolate the threat, prevent data loss, and preserve evidence:

Host Isolation

Compromised web server (192.168.5.200) quarantined using VLAN segmentation
Firewall blocks added for attacker IP 167.172.3.114 (both inbound & outbound)

Malicious File Removal

Web shells (eval-stdin.php, c99shell.php, shell.php) removed after forensic capture
Apache service stopped to disable web access temporarily

Perimeter Hardening

Egress filtering enforced – outbound traffic restricted to trusted domains
IDS rules updated with latest Emerging Threat signatures

Containment Strategy

Access Lockdown

Admin accounts disabled; MFA enforced for all employees

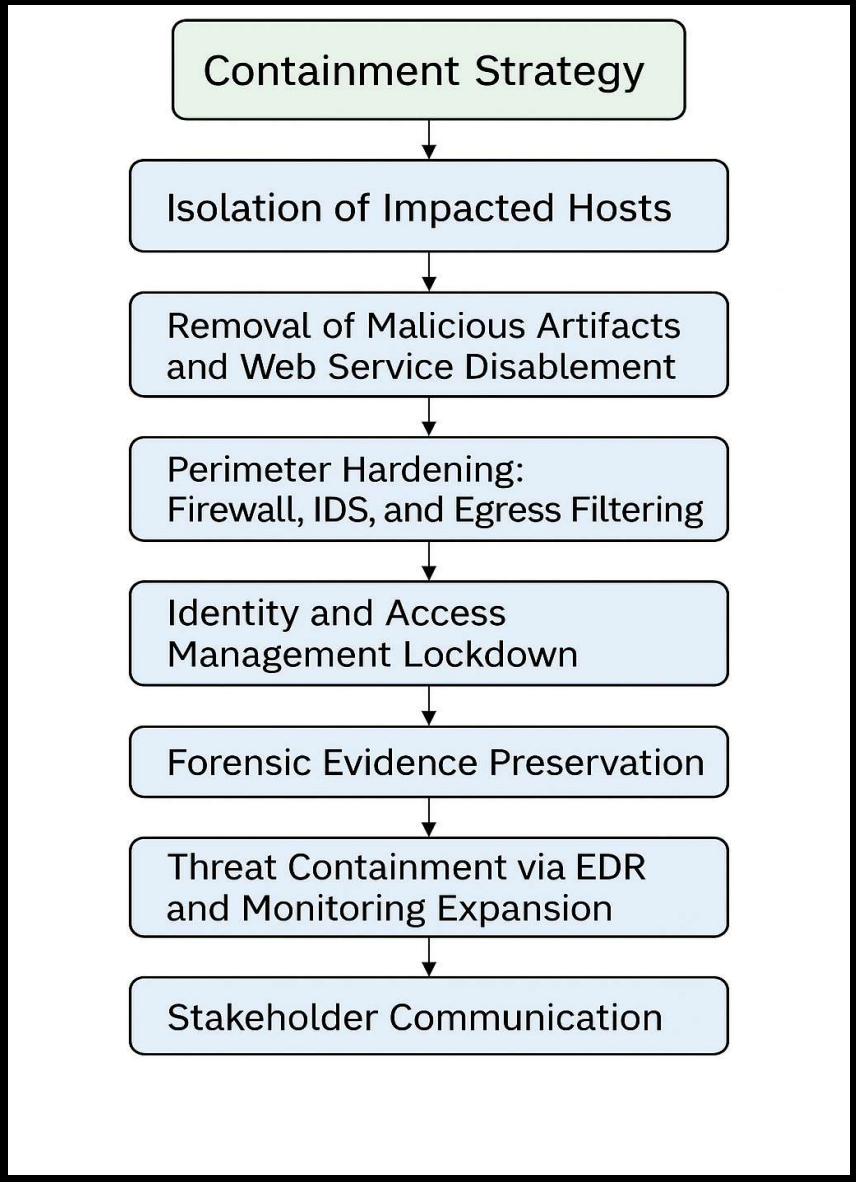
Reviewed SSH logs, sudo attempts, and suspicious shell history

Forensic Preservation

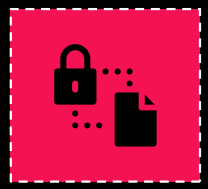
Logs archived and hashed (SHA256)
Full disk image captured using dd for deep analysis

Monitoring Expansion

Deployed Wazuh + Sysmon across DMZ and LAN
Additional sensors configured for enhanced visibility



Recovery Plan

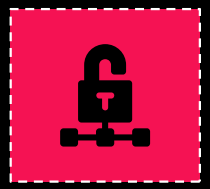


OS Reinstallation

Clean Ubuntu Server 22.04 LTS image

System Hardening

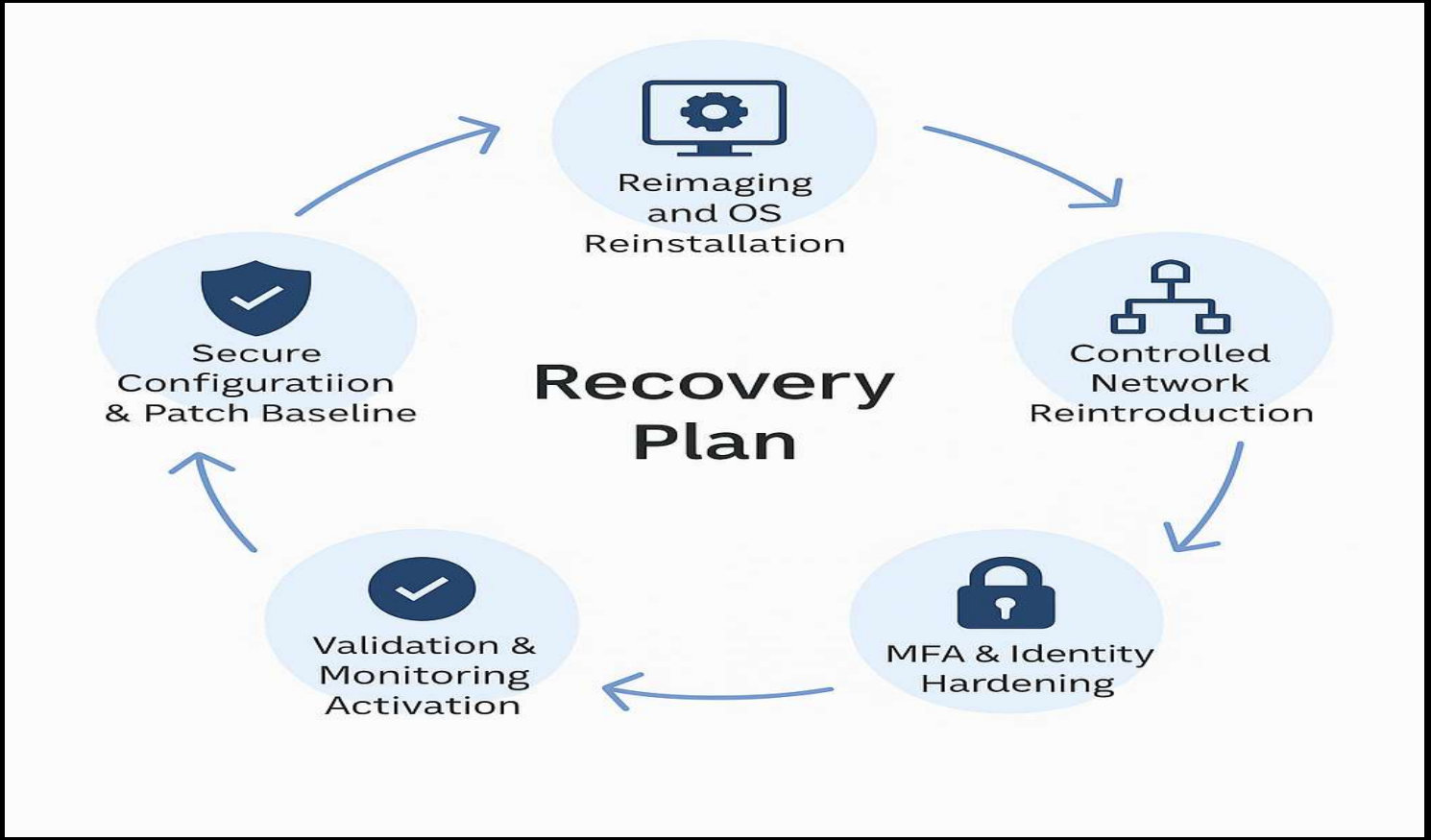
Patching, secure configurations



Reintegration Process

Monitored staging before production rollout

Recovery Plan



Post-Incident Review & Root Cause Analysis 01

- **Executive Summary**

The attacker exploited SQL Injection and Remote Code Execution (RCE) vulnerabilities
Breach lasted from April 10 to April 19, 2025
Over 50GB of data exfiltrated (backups, DB configs, CSVs)

- 🛠 **Root Causes Identified**

Technical:

Poor input sanitization in login.php
Dangerous PHP functions (system(), exec()) enabled
No Web Application Firewall (WAF), Network Firewall not configured properly

Post-Incident Review & Root Cause Analysis 01

- **Organizational:**

  Lack of secure SDLC practices

  No routine application security testing

  MFA not enforced for admin access

- **Security Controls Gaps:**

Control	Status During Attack	Impact
MFA	Disabled	Allowed lateral movement
EDR	Not installed	Shell activity undetected
WAF	Not deployed	RCE payloads reached server
PHP HARDENING	Weak	Enabled RCE via web shell
Egress Filtering	Not onfigured	Exfiltration went unnoticed
Centralized Logging	Partially effective	Alerts triggered post-exploit

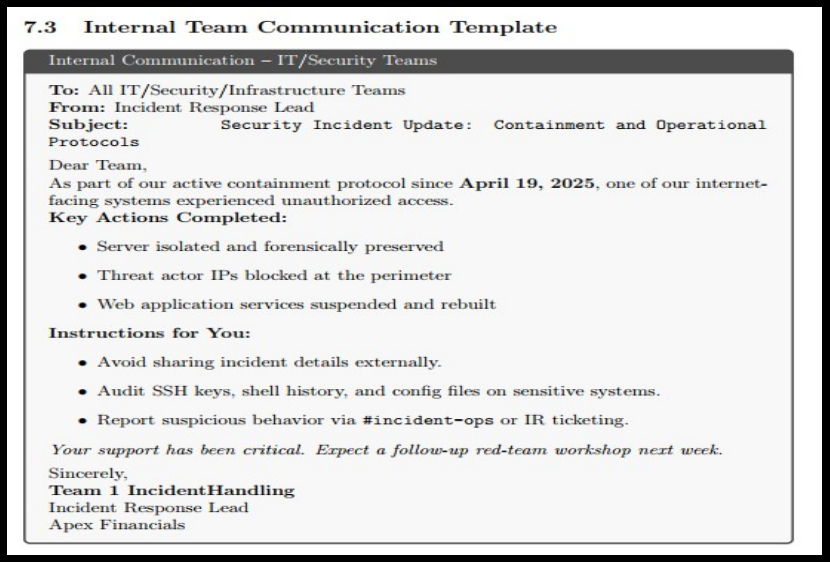
Stakeholder Communication Strategy

Executive Briefing: Scope of attack, response actions

Internal Team Update: Containment protocols, best practices

External Communication: Client notifications, improved security assurance

Stakeholder Communication Strategy



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:

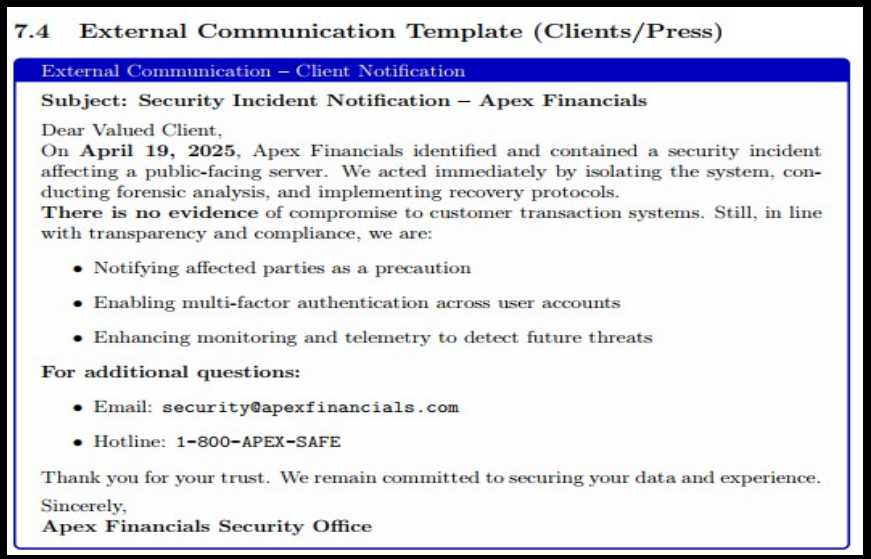
- Server isolated and forensically preserved
- Threat actor IPs blocked at the perimeter
- Web application services suspended and rebuilt

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

Executive Briefing

Scope of attack, response actions

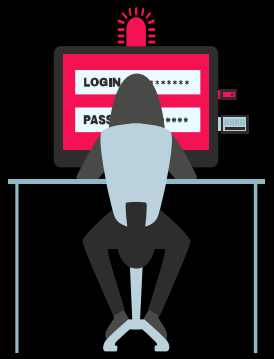
Internal Team Update

Containment protocols, best practices

External Communication

Client notifications, improved security assurance

Advanced Threat Intelligence & Bonus Findings



- **Objective:** Deepen attacker profiling and understand post-exploitation behavior

- **C2 Infrastructure Enumeration**
  Attacker IP: 167.172.3.114
  C2 Channel: Reverse HTTP POST from internal server (192.168.5.200)

- **Traits:**
  No DNS – direct IP-based connection
  Long POST sessions → consistent with reverse shell behavior

- **Exfiltrated Data Evidence**
  Estimated Volume: 50GB+
  Accessed Files: /etc/passwd, /var/www/html/db_config.php, /home/admin/backup.tar.gz

Advanced Threat Intelligence & Bonus Findings

- **File Types:**
  Configs, logs, .csv, .xls, compressed backups

- **Backdoor Capabilities – c99shell.php**

  Remote command execution (e.g., whoami, ls)
  File system browsing & uploads/downloads
  Reverse shell with built-in curl / nc
  Dynamic function calls to evade basic AV

- **MITRE ATT&CK Techniques Observed**

  T1059.003 – PHP Command Execution
  T1110 – Brute Force Login
  T1505.003 – Web Shell
  T1041 – Data Exfiltration via C2

Conclusion & Key Takeaways

- **Summary of the Incident**

  Web server compromised via SQL Injection and Remote Code Execution
  Attacker used c99shell.php to gain persistent access
  Over 50GB of sensitive data exfiltrated to external IP
  Logs analyzed from web server, IDS, and firewall using Splunk

- **Key Lessons Learned**

  MFA must be mandatory across all admin access
  Web Application Firewall (WAF) is critical for input filtering
  PHP hardening and secure coding practices are essential
  Centralized logging and real-time alerting should be fully deployed
  Red team simulations help prepare for real-world attack scenarios

- **Next Steps (Simulated Recommendations)**

  Deploy full EDR + WAF + SIEM stack
  Enforce secure SDLC with regular vulnerability scans
  Conduct security awareness training for all technical staff

THANKS!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**.