



Cybersecurity Incident Report - Apex Financials (INC250422)

Group 1



Guided by - Dr.Premila Melvin, John Griffin

Group Members

01 Leela Pavan

04 Shalem Raju

02 Phanindhar Reddy

05 Sriram

03 Junaid

06 Srivarsha



Table of Contents

- | | |
|--|---|
| 01
Executive Summary & Case Report | 06
Recovery Plan |
| 02
Root Cause Identification & Log Analysis | 07
Post-Incident Review & Root Cause Analysis |
| 03
Scope and Impact of the Incident | 08
Stakeholder Communication Strategy |
| 04
Reconstructed Attack Timeline | 09
Advanced Threat Intelligence & Bonus Findings |
| 05
Containment Strategy | 10
Conclusion & Key Takeaways |





Executive Summary

- 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

- 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

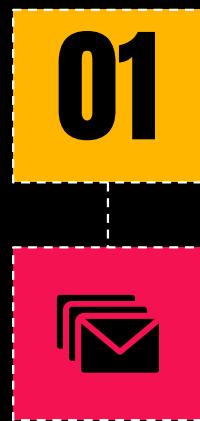
- Reconstructing the second-by-second attack timeline

- Identifying Indicators of Compromise (IoCs)

- Confirming the data exfiltration path

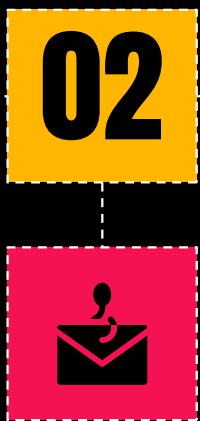


Attack Timeline



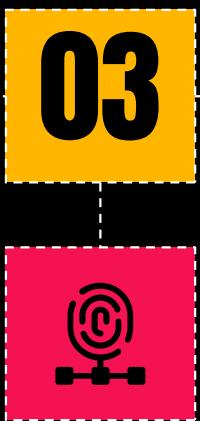
Reconnaissance

Enumeration of login page



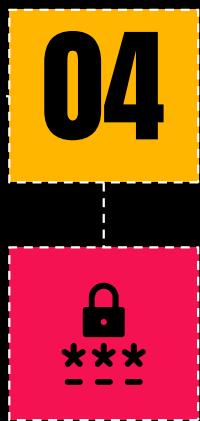
Exploitation

SQL injection, RCE via PHP shell



Privilege Escalation

Commands executed remotely



Data Exfiltration

50GB transferred via HTTP sessions

Root Cause Identification & Log Analysis



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

This screenshot shows a log analysis interface with several log entries. The logs are from a MySQL database and detail failed login attempts on the '/login.php' page. The logs include the IP address of the attacker, the timestamp, and the specific error message indicating a failed login attempt. The interface has various filters and search options at the top.

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)

This screenshot shows a log analysis interface with several log entries. The logs are from a MySQL database and detail command executions via the 'cmd=' parameter. These logs show the attacker running commands like 'ls' and 'whoami' to list files and check privileges. The interface has various filters and search options at the top.

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



New Search	
<code>index=main host=192.168.5.200 sourcetype=id_logs_pipe "167.172.3.114"</code>	
✓ 6 events (before 5/2/25 9:34:40,000 AM) No Event Sampling ▾	
Events (6) Patterns Statistics Visualization	
◀ Timeline format ▾	— Zoom Out ▾
Format ▾	Show 50 Per Page ▾
View List ▾	
Selected Fields	
host_1	192.168.5.200
source_1	id_logs_pipe
Interesting Fields	
date_hour_2	2025-04-10T17:25:30-0400
date_minute_2	2025-04-10T17:25:30-0400
date_second_2	2025-04-10T17:25:30-0400
date_hour_5	2025-04-10T17:25:30-0400
date_minute_5	2025-04-10T17:25:30-0400
date_second_5	2025-04-10T17:25:30-0400
date_year_1	2025-04-10T17:25:30-0400
date_time_1	2025-04-10T17:25:30-0400
destination_ip_2	167.172.3.114
index_6	id_logs_pipe
indexcount_1	1
logoffset_1	0
protocol_6	TCP
source_ip_2	192.168.5.200
signature_Alert_Name_6	IDS
signature_server_1	IDS
timestamp_1	2025-04-10T17:25:30-0400
timestamp_5	2025-04-10T17:25:30-0400
Extract New Fields	
Time Event	
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB_SERVER Possible Command Execution via Web Shell CRITICAL HTTP GET request with 'cmd' parameter detected in request for 'vshell1.php' from 167.172.3.114
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB_SERVER Possible Web Shell Access CRITICAL HTTP GET request for '/shell1.php' detected from 167.172.3.114
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB_SERVER Possible Web Shell Access CRITICAL HTTP GET request for '/shell1.php' detected from 167.172.3.114
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER SUSPICIOUS OUTBOUND HTTP POST - Potential Data Exfiltration HIGH Dubious POST request detected from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible SQL Injection Attempt: '< OR >' HIGH HTTP GET request with potential SQL injection syntax in parameters from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible SQL Injection Attempt: '< OR >' HIGH HTTP GET request with potential SQL injection syntax in parameters from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible Web Application Attack - Directory Traversal Attempt HIGH HTTP GET request for '/..;/etc/passwd' detected from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible Web Application Attack - Directory Traversal Attempt HIGH HTTP GET request for '/..;/etc/passwd' detected from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible Web Application Attack - Multiple Anomalies HIGH Multiple suspicious HTTP requests from 167.172.3.114 to 192.168.5.200
> 4/10/25 5:23:35,000 PM	2025-04-10T17:25:30-0400 167.172.3.114 192.168.5.200 TOP ET WEB SERVER Possible Web Application Attack - Multiple Anomalies HIGH Multiple suspicious HTTP requests from 167.172.3.114 to 192.168.5.200

splunk-enterprise Apps ▾	
Overview Summary Health Check Indexing Search Resource Usage Forwarders Settings Run a Search Monitoring Console	
Administrator Messages Settings Activity Help Find	
Save As Create Table View Close	
◀ Timeline format ▾	— Zoom Out ▾
Format ▾	Show 50 Per Page ▾
View List ▾	
New Search	
<code>index=main host=fw_gateway1 sourcetype=id_logs_pipe "192.168.5.200" "167.172.3.114" "ALLOWED"</code>	
✓ 3 events (before 5/2/25 11:04:00,000 PM) No Event Sampling ▾	
Events (3) Patterns Statistics Visualization	
◀ Timeline format ▾	— Zoom Out ▾
Format ▾	Show 50 Per Page ▾
View List ▾	
Selected Fields	
host_1	192.168.5.200
source_1	fw_logs_bt
Interesting Fields	
action_1	ALLOWED
bytesReceived_3	53687951200
bytesSent_3	6789
date_hour_1	2025-04-19T17:38:00-0400
date_minute_1	2025-04-19T17:38:00-0400
date_second_1	2025-04-19T17:38:00-0400
date_zone_1	2025-04-19T17:38:00-0400
destinationIP_Port_1	167.172.3.114:80
index_1	fw_logs_bt
inCount_1	1
protoCount_1	1
Time Event	
> 4/9/25 5:30:00,000 PM	2025-04-19T17:38:00-0400 192.168.5.200 167.172.3.114:80 TOP ALLOWED 53687951200 6789
> 4/9/25 5:30:00,000 PM	2025-04-19T17:38:00-0400 192.168.5.200 167.172.3.114:80 TOP ALLOWED 5678 123
> 4/9/25 5:30:00,000 PM	2025-04-19T17:38:00-0400 192.168.5.200 167.172.3.114:80 TOP ALLOWED 5678 123

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

Containment Strategy

Isolation of Impacted Hosts

Removal of Malicious Artifacts
and Web Service Disablement

Perimeter Hardening:
Firewall, IDS, and Egress Filtering

Identity and Access
Management Lockdown

Forensic Evidence Preservation

Threat Containment via EDR
and Monitoring Expansion

Stakeholder Communication

Recovery Plan



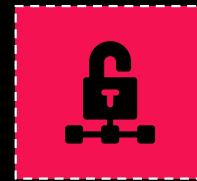
OS Reinstallation

Clean Ubuntu Server 22.04 LTS image



System Hardening

Patching, secure configurations

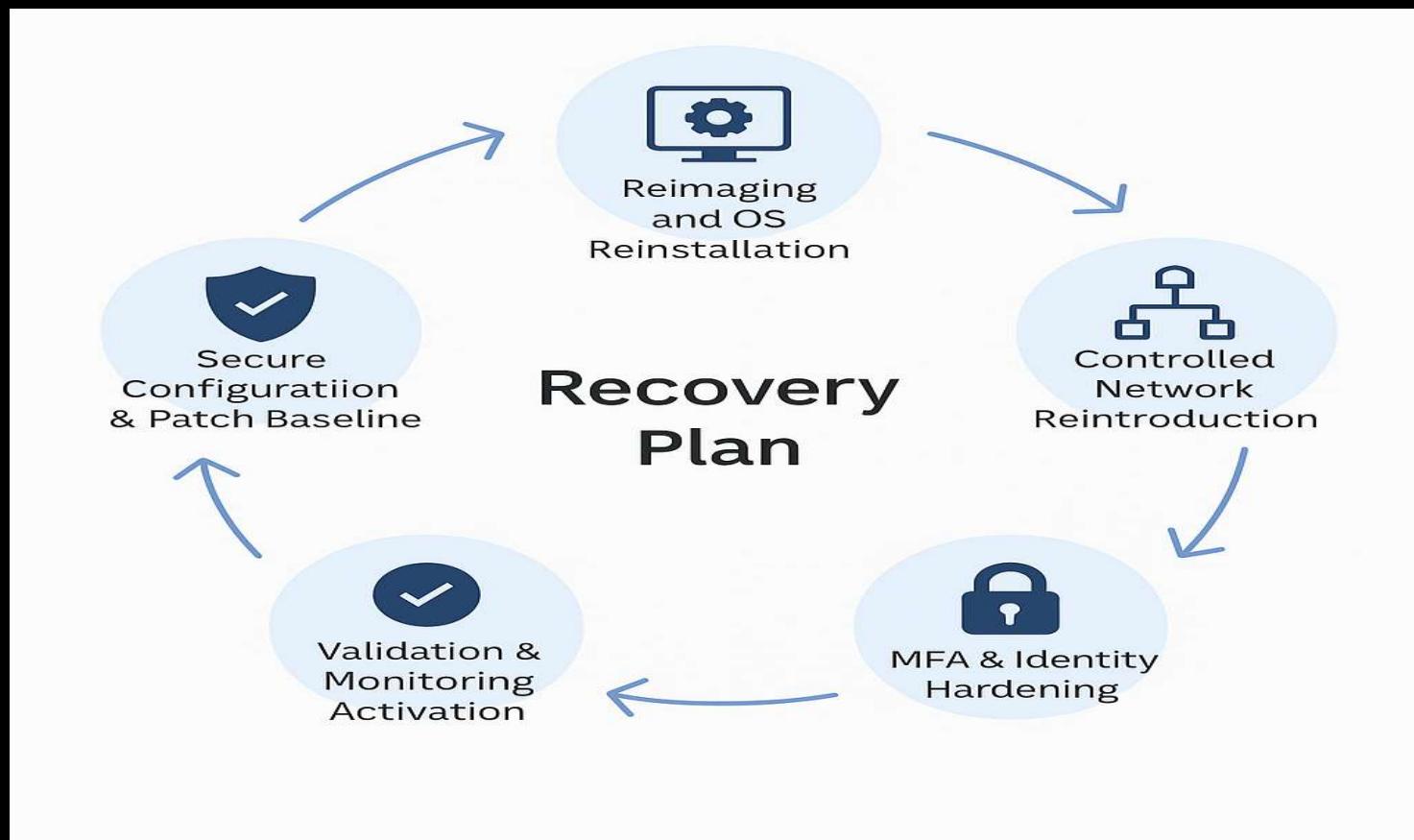


Reintegration Process

Monitored staging before production rollout



Recovery Plan



01

Post-Incident Review & Root Cause Analysis

- 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

- **Organizational:**

- Lack of secure SDLC practices
- No routine application security testing
- MFA not enforced for admin access

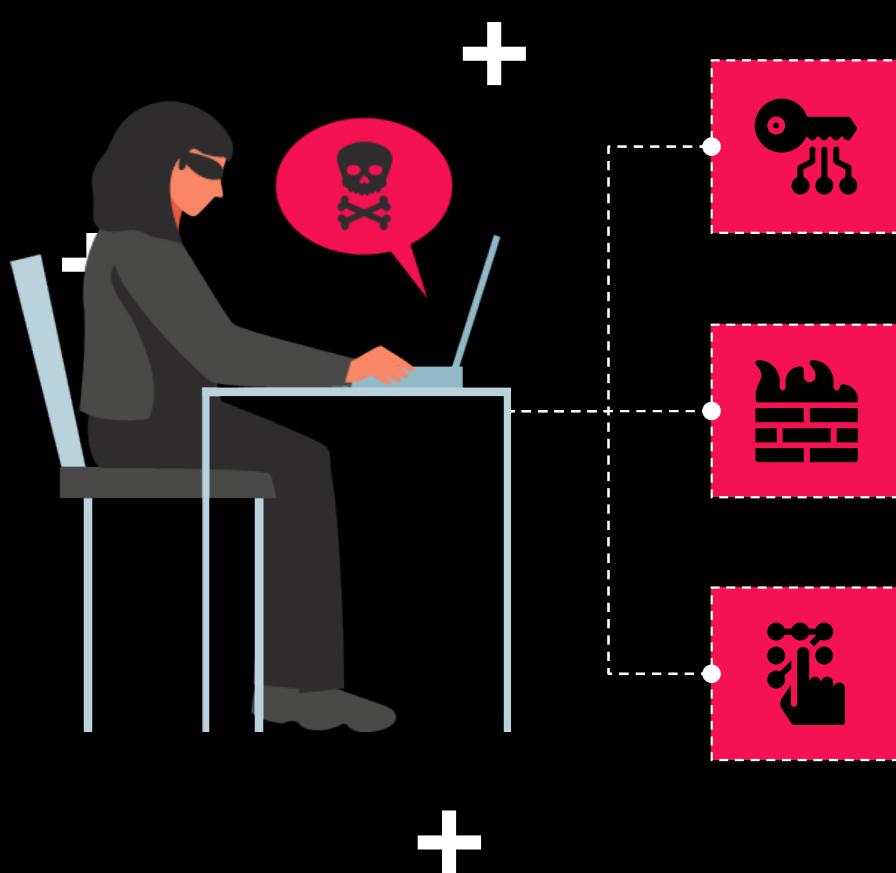


- **Security Controls Gaps:**

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 configured	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 Incident Handling
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](#).