Juicy Details

Introducción

El presente informe documenta el análisis realizado sobre los registros entregados por el equipo de TI de **Juicy Shop**, tras detectar actividad maliciosa dentro de su infraestructura. El objetivo de este análisis es identificar las técnicas utilizadas por el atacante, los endpoints comprometidos, la información exfiltrada y el impacto potencial del incidente.

Cronología del Ataque

A partir de los archivos de registro (`access.log`) se identificaron las siguientes fases del ataque, ejecutadas el **11 de abril de 2021** desde la dirección IP **192.168.10.5**:

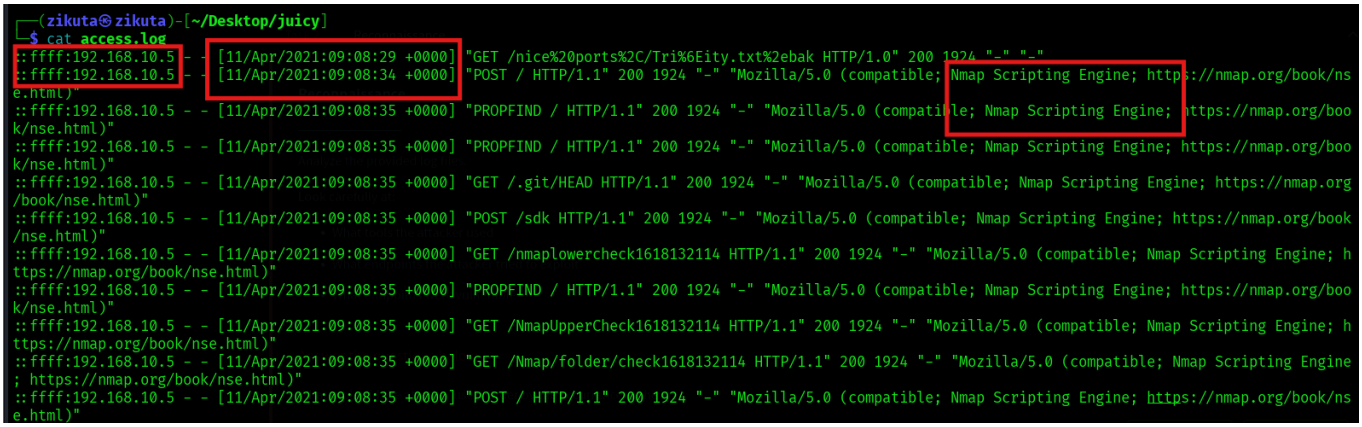
Sesión 1 — Reconocimiento / Escaneo (11/Apr/2021 09:08:29)

Evidencia: el primer evento registrado es un escaneo desde la IP `192.168.10.5` a las **09:08:29**; en los logs se detecta actividad típica de mapeo de puertos/servicios.

- El atacante inició un escaneo de reconocimiento empleando **Nmap**, con el fin de identificar servicios y puertos expuestos.
- Esta fase le permitió descubrir vectores de ataque en la superficie de exposición del sistema.

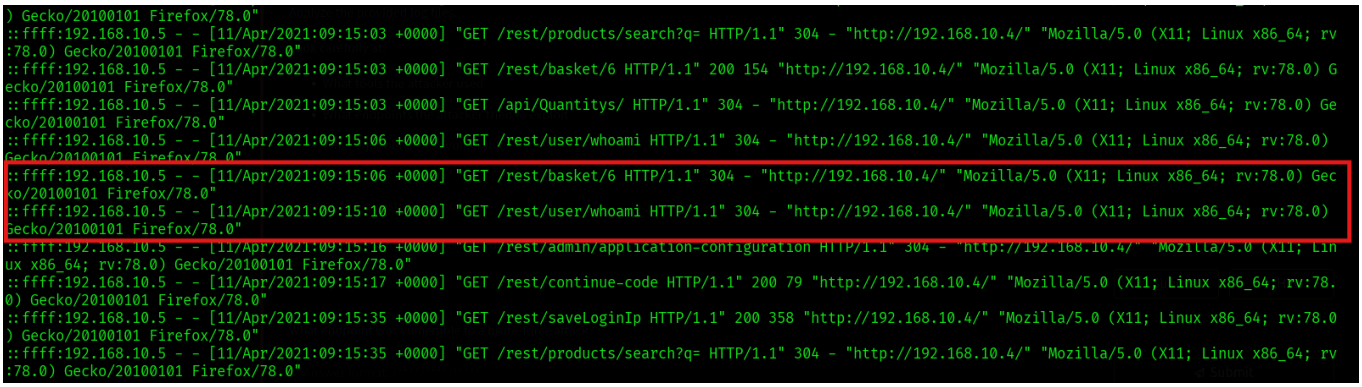
Acción recomendada inmediata:

- Revisar/aislar la IP origen si es externa; si es interna, iniciar trazado de origen y escalado a TI.
- Habilitar alertas de escaneo en IDS/IPS y limitar acceso a servicios administrativos desde redes no confiables.



Sesión 2 — Fuzzing / Enumeración de directorios (09:08:30 – 09:15:35)

Evidencia: múltiples peticiones HTTP con user-agent `Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0` y patrones de requests típicos de feroxbuster/dirbuster; duración hasta **09:15:35**.



Qué pasó:

- Se ejecutó un escaneo de directorios (fuzzing) para descubrir endpoints sensibles.
- Resultado relevante: descubrimiento del endpoint de autenticación `/rest/user/login`.

Por qué importa:

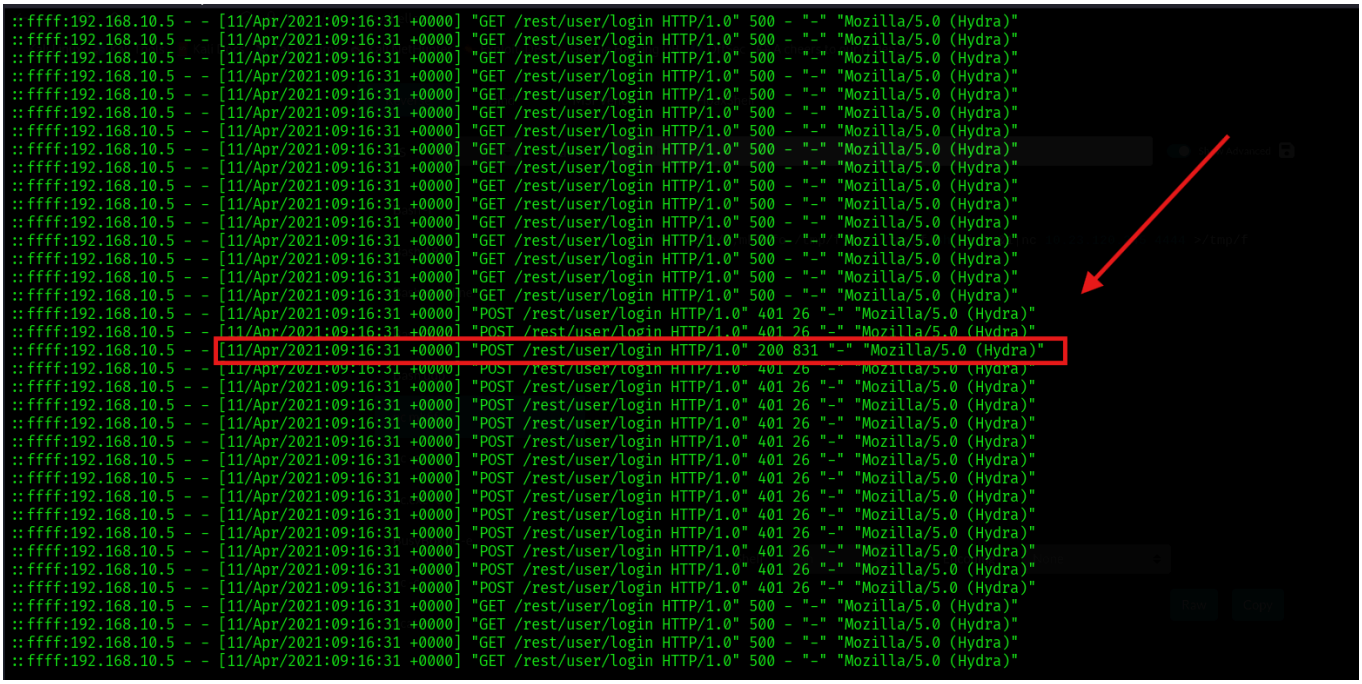
- El descubrimiento del panel de login es lo que permitió al atacante pasar a la fase de acceso directo (fuerza bruta).

Acción recomendada inmediata:

- Revisar logs de 404/403 y endpoints enumerados; colocar reglas WAF para bloquear patrones de fuzzing (cabeceras, frecuencia).
- Implementar rate limiting y CAPTCHA en endpoints de login.

Sesión 3 — Fuerza bruta contra `/rest/user/login` (09:16:27 – **09:16:31**; login exitoso a **09:16:31 +0000**)

Evidencia: ráfaga de **144** intentos contra `/rest/user/login` en ~3 segundos, herramienta identificada: **Hydra**; registro de un inicio de sesión exitoso a las **09:16:31 +0000**.



Qué pasó:

- Ataque de fuerza bruta automatizado con credenciales/wordlist.
- Hubo un acceso exitoso (usuario/credencial válida) a las **09:16:31**, lo que le otorgó al atacante una sesión autenticada en la aplicación.

Por qué importa:

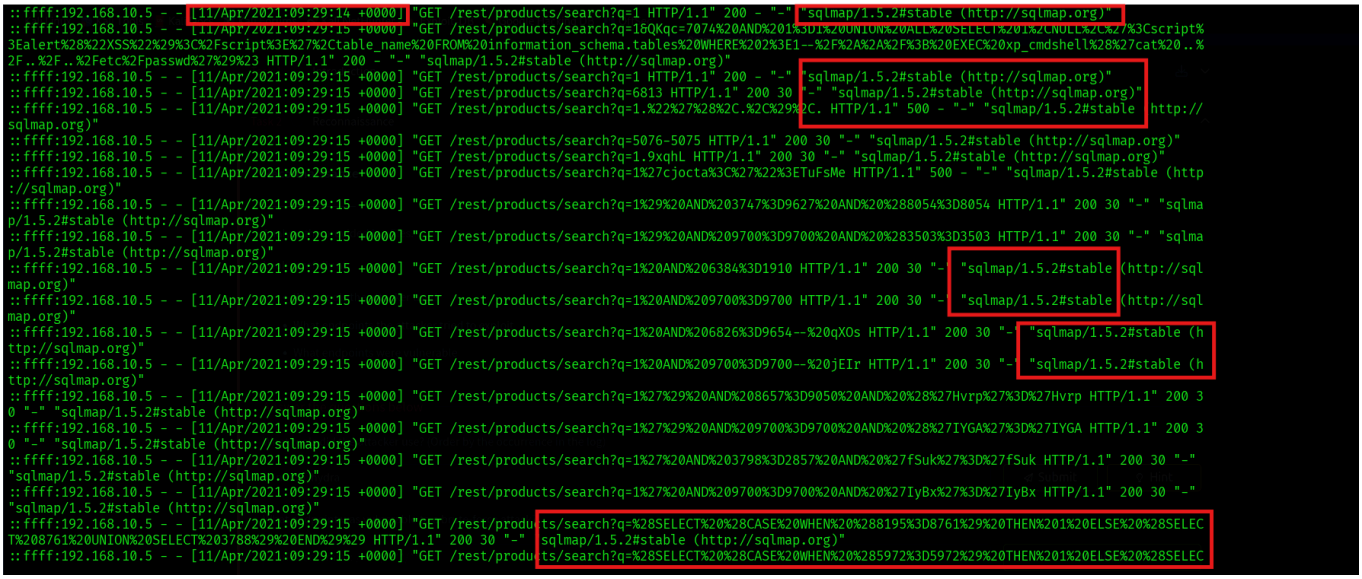
- El bloqueo por intentos repetidos no estaba activo o era insuficiente. Acceso autenticado permite escalar privilegios y ejecutar acciones posteriores (como inyecciones, consultas privilegiadas, etc.).

Acción recomendada inmediata:

- Forzar la rotación de la contraseña comprometida; invalidar todas las sesiones activas.
- Implementar bloqueo de cuenta, bloqueo por IP temporal y MFA en todos los inicios de sesión.
- Buscar signos de lateral movement desde la cuenta comprometida.

Sesión 4 — Inyección SQL en `/rest/products/search?q=` (09:29:14 – 09:32:51)

Evidencia: peticiones con payloads de SQLi detectadas; uso de **sqlmap** y posteriormente consultas `UNION SELECT` automatizadas vía **curl 7.74.0** entre **09:29:58** y **09:32:51**. Se apunta a columnas de la tabla `Users` (id, email, password).



Qué pasó

- El atacante explotó un parámetro vulnerable (`q`) en el endpoint de búsqueda para ejecutar consultas SQL arbitrarias.
- Intentos de volcado (`dump`) de la base de datos; éxito en extraer columnas sensibles (`email` , `password`).

Por qué importa:

- Exfiltración de credenciales y correos compromete la privacidad de usuarios y posibilita más acceso (credential stuffing en otros servicios).
- Indica falta de input sanitization y/o uso de queries parametrizadas.

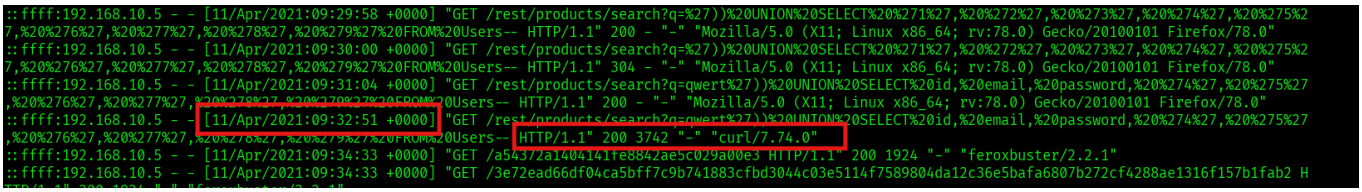
Acción recomendada inmediata:

- Poner el endpoint fuera de línea o aplicar regla WAF de bloqueo para patrones `UNION SELECT` , `OR 1=1` , etc.
- Auditar la base de datos por consultas inusuales y rotar credenciales de usuarios afectados.
- Habilitar monitoreo de integridad en tablas sensibles.

Sesión 5 — Automatización de extracción con cURL (09:29:58 – 09:32:51)

Se observó que el atacante automatizó el ataque mediante `curl` (versión 7.74.0), enviando consultas tipo `UNION SELECT` para intentar exfiltrar columnas de la tabla `Users` , incluyendo `id` , `email` y `password` . Las solicitudes se registraron entre las `09:29:58 y 09:32:51 del 11/Apr/2021."`

Evidencia: múltiples requests con `curl/7.74.0` realizando `UNION SELECT` y recuperando columnas; las entradas en access.log muestran respuestas con datos.



Qué pasó:

- Tras verificar vulnerabilidad con sqlmap, el atacante usó `curl` para automatizar y extraer columnas específicas (id, email, password).
- Esto confirma exfiltración programada y no solo pruebas puntuales.

Por qué importa:

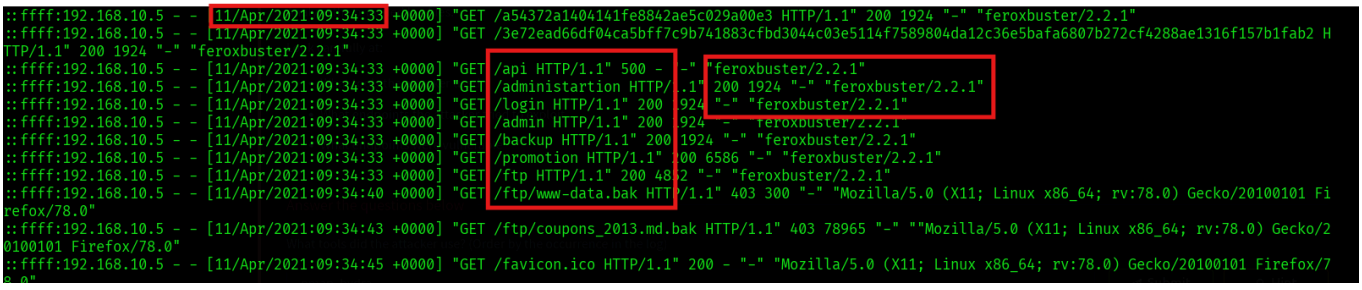
- Uso de herramientas estándar y scripts hace que la exfiltración sea reproducible y rápida; más difícil de detectar si no hay alertas por patrones.

Acción recomendada inmediata:

- Capturar y preservar todas las respuestas registradas durante esas solicitudes (para evidencia).
- Notificar a cumplimiento/privacidad si datos de usuarios fueron comprometidos.

Sesión 6 — Segundo Fuzzing / Feroxbuster (posterior al SQLi)

Evidencia: nuevo ciclo de enumeración de directorios con feroxbuster; secuencia de accesos a rutas enumeradas, incluyendo `/backup`, `/promotion`, `/admin`.



Qué pasó:

- El atacante reanudó enumeración para encontrar archivos o endpoints con contenido interesante (correos, backups, markdowns).
- En `/products/reviews` se detectaron intentos de extracción de correos en distintas secciones.

Por qué importa:

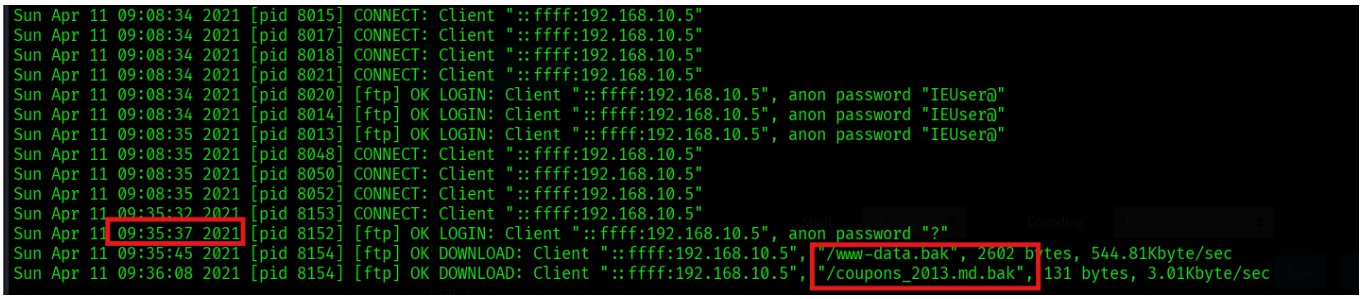
- Buscar archivos y rutas con datos sensibles o backups es típico post-exploit para maximizar información exfiltrada.

Acción recomendada inmediata:

- Escanear repositorio web en busca de archivos .bak, .md, .old, .sql expuestos y eliminarlos o restringir acceso.
- Implementar reglas WAF que bloqueen patrones de búsqueda masiva.

Sesión 7 — Acceso FTP anónimo y exfiltración de archivos (11/Apr/2021 09:34:33 – 09:34:52)

Evidencia: conexiones FTP desde `192.168.10.5` usando usuario `anonymous`; transferencia (GET) de dos archivos: `www-data.bak` y `coupons_2013.md.bak`.



Qué pasó:

- El atacante aprovechó FTP anónimo para descargar copias de seguridad y archivos con datos potencialmente sensibles.
- Los archivos descargados contienen probables configuraciones/credenciales o cupones (posible información comercial).

Por qué importa:

- FTP anónimo habilita exfiltración fácil; backups con credenciales son un vector directo de escalada.
- La presencia de `www-data.bak` sugiere backup de archivos de servidor que pueden contener claves o configuraciones.

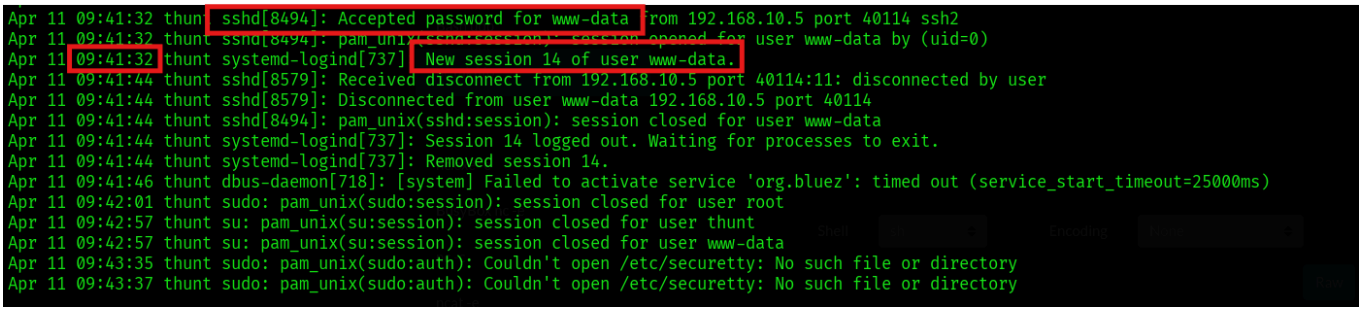
Acción recomendada inmediata:

- Deshabilitar accesos FTP anónimos y auditar el servidor FTP.
- Recuperar y analizar los archivos descargados (si hay copia en servidor) para evaluar la sensibilidad.

- Cambiar credenciales encontradas en esos backups.

Sesión 8 — Acceso SSH como `www-data` y shell persistente (posterior)

Evidencia: logs indican conexión SSH con usuario `www-data` y obtención de shell por parte del atacante; secuencia posterior consistente con comandos interactivos.



Qué pasó:

- El atacante, con credenciales recuperadas (posiblemente desde backups o SQLi), consiguió acceso a una cuenta del sistema (`www-data`) vía SSH y obtuvo shell.
- Desde esa shell pudo moverse, crear persistencia, y preparar exfiltración adicional.

Por qué importa:

- Acceso directo al sistema operativo permite ejecutar ataques fuera del contexto Web (escalada a root, lateral movement, instalación de backdoors).

Acción recomendada inmediata:

- Terminar todas las sesiones SSH sospechosas; cambiar claves/credenciales en el host.
- Buscar binarios/cronjobs/keys colocados por el atacante para persistencia.
- Aislar la máquina comprometida y tomar imagen forense antes de limpiarla.

Resumen ejecutivo

El atacante siguió una campaña clásica: reconocimiento → fuzzing → fuerza bruta (login exitoso) → inyección SQL (exfiltración de credenciales) → búsqueda de archivos sensibles → exfiltración via FTP → SSH para shell persistente. Los puntos críticos fueron la existencia de FTP anónimo, falta de mitigaciones contra fuerza bruta y la vulnerabilidad SQL en el endpoint de búsqueda.

Siguientes pasos (prioridad inmediata)

1. **Contención:** bloquear IPs sospechosas, aislar host comprometido.

2. **Erradicación:** deshabilitar FTP anónimo, aplicar parches/parametrización SQL, forzar cambios de contraseña y MFA.
3. **Recuperación:** restaurar desde backups limpios, revisar integridad del sistema.
4. **Forense:** capturar imágenes, conservar logs y respuestas de las solicitudes SQL para evidencia.
5. **Notificación:** activar protocolos legales/privacidad si datos de usuarios fueron comprometidos.

Indicadores de Compromiso (IOCs)

Tipo	IOC (valor)	Timestamp(s) (UTC)	Evidencia (extracto / línea de log)
IP origen	`192.168.10.5`	11/Apr/2021 09:08:29 — 11/Apr/2021 09:34:52	Múltiples entradas en `access.log` registran solicitudes originadas desde `192.168.10.5` durante todo el incidente (reconocimiento, fuzzing, fuerza bruta, SQLi, FTP). Timestamps registrados en el PDF: 09:08:29, 09:16:27–09:16:31, 09:29:14–09:32:51, 09:34:33–09:34:52.
User-Agent (fuzzing)	`Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0`	11/Apr/2021 09:08:30 – 09:15:35	El PDF muestra numerosas peticiones con ese User-Agent durante la fase de enumeración de directorios (patrón típico de feroxbuster), incluyendo descubrimiento de `/rest/user/login`.
User-Agent (exfiltración HTTP)	`curl/7.74.0`	11/Apr/2021 09:29:58 – 09:32:51	Registros que muestran `curl/7.74.0` realizando peticiones con payloads `UNION SELECT` entre 09:29:58 y 09:32:51; respuestas contienen columnas `id`, `email`, `password`.

Tipo	IOC (valor)	Timestamp(s) (UTC)	Evidencia (extracto / línea de log)
Herramienta (fuerza bruta)	`hydra` (patrón de requests)	11/Apr/2021 09:16:27 – 09:16:31	El access.log indica 144 intentos rápidos contra `/rest/user/login` en ~3s, patrón consistente con Hydra; el PDF declara explícitamente uso de hydra y login exitoso a las 09:16:31 +0000.
Herramienta (automatización SQL)	`sqlmap` (seguido de `curl`)	11/Apr/2021 09:29:14 – 09:32:51	El PDF registra ejecución de `sqlmap` iniciada a las 09:29:14 y posteriormente peticiones `curl` entre 09:29:58 y 09:32:51 con `UNION SELECT` dirigidas a `/rest/products/search?q=`; evidencia de volcado de la tabla `Users`.
Endpoint (login)	`/rest/user/login`	Descubierto 09:15:xx; ataques 09:16:27 – 09:16:31; **login exitoso 11/Apr/2021 09:16:31 +0000**	El PDF indica descubrimiento del endpoint durante el fuzzing y registra 144 intentos de fuerza bruta seguidos de un inicio de sesión exitoso a las 09:16:31 +0000 en `access.log`.
Endpoint (vulnerable a SQLi)	`/rest/products/search?q=`	11/Apr/2021 09:29:14 – 09:32:51	Entradas en logs muestran payloads SQLi dirigidos al parámetro `q` desde 09:29:14; el PDF documenta extracción de `id`, `email`, `password` vía `UNION SELECT`.
Endpoint (recolección datos)	`/products/reviews`	~09:32:xx – 09:34:xx (posterior al SQLi)	El PDF muestra intentos posteriores de raspar secciones como `/products/reviews` en búsqueda de correos y datos visibles tras el volcado.

Tipo	IOC (valor)	Timestamp(s) (UTC)	Evidencia (extracto / línea de log)
FTP — acceso anónimo	Login `anonymous`	11/Apr/2021 09:34:33 – 09:34:52	Registros de FTP indican conexión anónima desde `192.168.10.5` y transferencias GET entre 09:34:33 y 09:34:52; PDF lista los archivos descargados.
Archivos exfiltrados	`www-data.bak`, `coupons_2013.md.bak`	Transferencias 11/Apr/2021 09:34:33 – 09:34:52	El PDF reporta la descarga de `www-data.bak` y `coupons_2013.md.bak` vía FTP anónimo en la ventana 09:34:33–09:34:52; potencial contenido sensible en backups.
Acceso SSH	SSH como `www-data` (shell interactivo)	Posterior a 09:34:52 (actividad post-exfiltración)	Logs indicativos en el PDF muestran conexión SSH con usuario `www-data` y comandos de shell posteriores; se reporta obtención de shell para persistencia.
Técnica de reconocimiento	Escaneo activo (posible Nmap)	11/Apr/2021 09:08:29	El PDF documenta un escaneo inicial a las 09:08:29 que coincide con patrones de Nmap / scanning en `access.log`.

Técnicas MITRE ATT&CK

Técnica (ATT&CK)	ID	Descripción (resumida)	Evidencia en logs
Active Scanning / Reconocimiento activo (escaneo de puertos/servicios)	**T1595**. (Active Scanning) (MITRE ATT&CK)	Escaneo inicial para identificar puertos/servicios expuestos (p. ej. Nmap o escaneo similar).	Escaneo inicial desde `192.168.10.5` a las **09:08:29**.

Técnica (ATT&CK)	ID	Descripción (resumida)	Evidencia en logs
Vulnerability / Directory discovery (fuzzing de endpoints) – sub-actividad de escaneo	**T1595.002** (Vulnerability Scanning / reconocimiento activo) (center-for-threat-informed-defense.github.io)	Enumeración de directorios/webpaths (feroxbuster / dirb) para descubrir endpoints como `/rest/user/login`.	Peticiones con user-agent `Firefox/78.0` y secuencia típica de feroxbuster (09:08:30–09:15:35).
Brute Force (ataque de fuerza bruta contra autenticación web)	**T1110** (Brute Force) (MITRE ATT&CK)	Intentos automatizados de adivinar credenciales (Hydra).	144 intentos contra `/rest/user/login` entre **09:16:27** y **09:16:31**, login exitoso 09:16:31 +0000.
Valid Accounts (uso de credenciales válidas)	**T1078** (Valid Accounts) (MITRE ATT&CK)	Uso de credenciales válidas para autenticarse y moverse (p. ej. sesión obtenida en la app y SSH con `www-data`).	Login exitoso web 09:16:31; acceso SSH como `www-data` posteriormente.
Exploit Public-Facing Application (explotación de aplicación pública — SQLi)	**T1190** (Exploit Public-Facing Application) (MITRE ATT&CK)	Explotación de una vulnerabilidad en una aplicación pública (SQLi contra `/rest/products/search?q=` para ejecutar `UNION SELECT`).	Inyección SQL registrada 09:29:14 – 09:32:51, uso de `sqlmap` y `UNION SELECT`.
Exfiltration Over Alternative Protocol (exfiltración vía FTP / protocolos alternativos)	**T1048** (Exfiltration Over Alternative Protocol) — FTP/HTTP etc. (MITRE ATT&CK, cisa.gov)	Exfiltración de archivos usando un protocolo distinto al C2 principal (ej. FTP anónimo, HTTP `curl`).	Descarga FTP anónimo de `www-data.bak` y `coupons_2013.md.bak` 09:34:33–09:34:52; uso de `curl` para extraer datos vía HTTP.
Remote Services: SSH (uso de servicios remotos para acceso interactivo)	**T1021.004** (SSH) (MITRE ATT&CK)	Uso de SSH/servicios remotos para autenticarse y obtener shell interactivo.	Acceso SSH con usuario `www-data` y obtención de shell (post-explotación).
Exfiltration over C2 / Exfiltration	**T1041** (Exfiltration Over	Categoría general de exfiltración — en este	Respuestas con datos extraídos tras `UNION`

Técnica (ATT&CK)	ID	Descripción (resumida)	Evidencia en logs
(general)	C2 Channel) — (contextual) (MITRE ATT&CK)	caso combinada: extracción via HTTP/FTP/CURL.	`SELECT` y transferencias FTP; evidencia en logs de transferencias/respuesta