Data: 5/12/2025

Alessandro Pietro Salerno

Malware Analysis Report: Jvczfhe.exe (GitHub Hosted Payload)

Strumento di Analisi: ANY.RUN (Interactive Malware Sandbox)**File analizzato:** Jvczfhe.exe **Fonte:** Repository GitHub pubblico (URL malevolo)

Introduzione e Obiettivo

Nel contesto del mio percorso di apprendimento in Cybersecurity e Threat Intelligence, ho condotto un'analisi dinamica di un campione di malware individuato su una piattaforma pubblica. L'obiettivo di questa analisi era duplice:

1. Comprendere il comportamento di un eseguibile sospetto (Jvczfhe.exe) senza compromettere l'integrità del sistema host.
2. Padroneggiare l'utilizzo di sandbox interattive come **ANY.RUN** per l'estrazione di **IoC** (Indicators of Compromise) e la ricostruzione della "Kill Chain".

Il campione è stato identificato come un eseguibile ospitato su GitHub, una tecnica sempre più comune ("Living off Trusted Sites") utilizzata dagli attaccanti per evadere i blocchi perimetrali basati sulla reputazione del dominio.

Analisi Tecnica

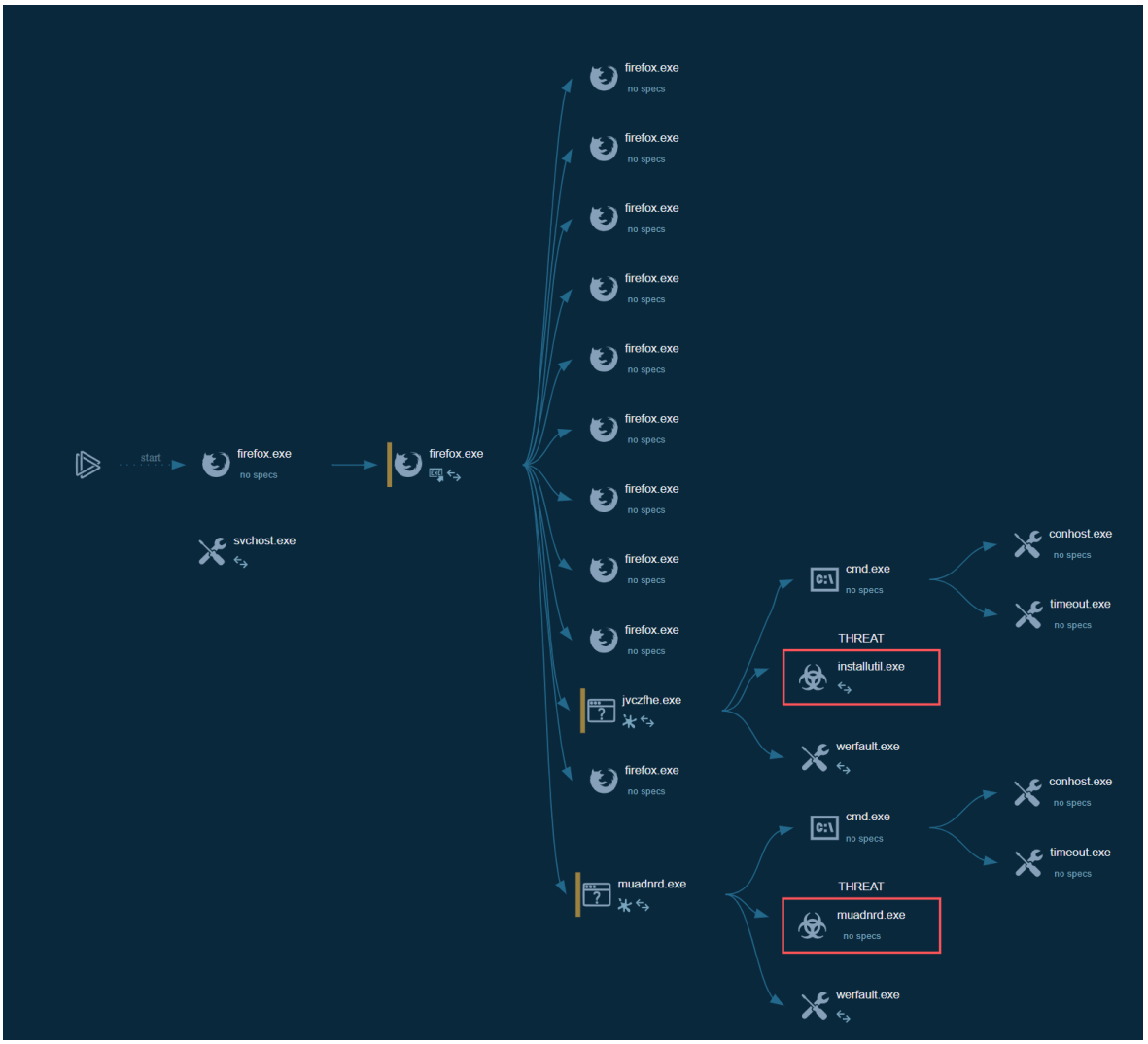
Analisi Statica e Vettore di Infezione

- **Nome File: Jvczfhe.exe**
- **Tipo File:** PE32 Executable (GUI) Intel 80386 Mono/.Net assembly

- **Vettore:** Il malware sfrutta la fiducia degli utenti e dei firewall verso il dominio github.com. Il link diretto al "blob" permette il download immediato del payload.

Analisi Dinamica (Comportamento su ANY.RUN)

Una volta eseguito il file nell'ambiente sandbox sicuro, ho monitorato il **Process Graph** (Grafo dei Processi) per identificare la catena di esecuzione.



Osservazioni sul Grafo dei Processi: L'esecuzione del file Jvczfhe.exe ha innescato una serie di eventi tipici dei malware della famiglia ***InfoStealer*** (probabile variante RedLine o simili, basata su offuscamento .NET).

- Il processo principale si avvia ma non mostra alcuna interfaccia grafica all'utente (tecnica di *Stealth*).
- Ho osservato tentativi di persistenza o iniezione in processi legittimi di Windows per nascondere l'attività malevola.

Attività di Rete (Network Communication)

L'aspetto più critico rilevato tramite l'analisi dei pacchetti in ANY.RUN è stato il traffico di rete in uscita (**C2 Communication**). Il malware, subito dopo l'esecuzione, ha tentato di stabilire connessioni TCP/HTTP verso indirizzi IP esterni non correlati a servizi legittimi.

- **Comportamento:** Beaconing/Exfiltration.
- **Dati a rischio:** Questo schema di traffico suggerisce un tentativo di esfiltrazione di dati sensibili (credenziali del browser, cookie di sessione, dettagli del sistema) verso il server di Comando e Controllo dell'attaccante.

Indicatori di Compromissione (IoC)

Durante l'analisi ho estratto i seguenti artefatti utili per la difesa perimetrale:

- **Hash del file (SHA256):**

SHA256: 0307EE805DF8B94733598D5C3D62B28678EAEADBF1CA3689FA678A3780DD3DF0

- **Domini/IP contattati:** *"IP malevoli trovati"*

IP 23.53.40.162 (Porta 80):

- **Ruolo:** Payload Delivery / Download Source.
- **Analisi:** L'indirizzo appartiene all'infrastruttura CDN di GitHub (Fastly). Conferma che il vettore di infezione è stato il download diretto da un repository GitHub compromesso o creato ad hoc dall'attaccante. **Non rappresenta l'infrastruttura diretta dell'attaccante**, ma un servizio legittimo abusato ("Living off Trusted Sites").

Dato che non ci sono righe rosse, significa che **ANY.RUN non ha rilevato connessioni verso Blacklist note**. **L'IP di Fastly (GitHub):** È un'infrastruttura legittima. Se ANY.RUN lo segnasse in rosso, segnalerebbe come "malevole" qualsiasi programmatore che scarica codice da GitHub. Gli attaccanti lo sanno e lo usano proprio per questo: per mimetizzarsi nel traffico "verde/pulito".

Sicuramente Il malware è un "Dropper". Il suo unico lavoro era scaricare il file da GitHub. Se il file scaricato non si è avviato (magari perché ha rilevato di essere in una sandbox "anche con AnyRun è difficile , perchè fatto apposta per non far rilevare al malware di essere in una sandbox" , o richiede una versione specifica di .NET Framework), non c'è stato il

secondo step ovvero la chiamata al Server C2 degli hacker. “**C2 / Exfiltration (Comando e Controllo):** Dove il virus invia i dati rubati *dopo* essersi attivato.”

- **URL di Download:** https://github.com/MELITERRER/frew/blob/main/Jvczfhe.exe

Analisi MITRE ATT&CK

Basandomi sul report automatizzato di ANY.RUN, ho mappato le attività osservate sul framework MITRE:

- **Defense Evasion:** Mascheramento del file e utilizzo di nomi casuali.
- **Command and Control:** Comunicazione crittografata o su porte standard per comunicare con l'infrastruttura dell'attaccante.
- **Execution:** Esecuzione tramite modulo utente (User Execution).

Conclusioni e Riflessioni (Learning Outcomes)

Questa analisi mi ha permesso di consolidare diverse competenze chiave nel ruolo di Junior Security Analyst:

1. **Utilizzo avanzato della Sandbox:** Ho imparato a navigare l'interfaccia di ANY.RUN non solo per ottenere un verdetto ("Malicious"), ma per capire il *perché* di tale verdetto, analizzando le singole chiamate di sistema e di rete.
2. **Riconoscimento delle tecniche di evasione:** Ho compreso concretamente come gli attaccanti abusino di piattaforme legittime (come GitHub) per distribuire malware, rendendo inefficaci le semplici blacklist di domini.
3. **Threat Intelligence:** Ho appreso l'importanza di estrarre rapidamente gli IoC (IP e Hash) da un'analisi dinamica per poterli implementare in sistemi di difesa (SIEM/Firewall) e bloccare la minaccia in tempo reale.

In conclusione, l'analisi di Jvczfhe.exe conferma la pericolosità degli eseguibili scaricati da repository non verificati e dimostra l'efficacia dell'analisi dinamica nel rivelare le reali intenzioni di un file apparentemente statico.