Analisi di Indicatori di Compromissione (Ioc) tramite anyrun

Introduzione

In questa sessione operativa, ho condotto un'analisi dinamica su un potenziale campione malware utilizzando la piattaforma di sandboxing anyrun. Lo scopo di questo Studio IoC è osservare il comportamento del malware in un ambiente isolato, tracciare la sua catena di esecuzione (Process Tree) e raccogliere artefatti tecnici (IP, domini, hash, modifiche di sistema) da inserire nelle difese perimetrali (Firewall, EDR, SIEM). Come best practice l'analisi dei risultati è stata visualizzata da una macchina virtuale ospitata su VirtualBox, garantendo un ulteriore livello di sicurezza.

Strumenti Utilizzati

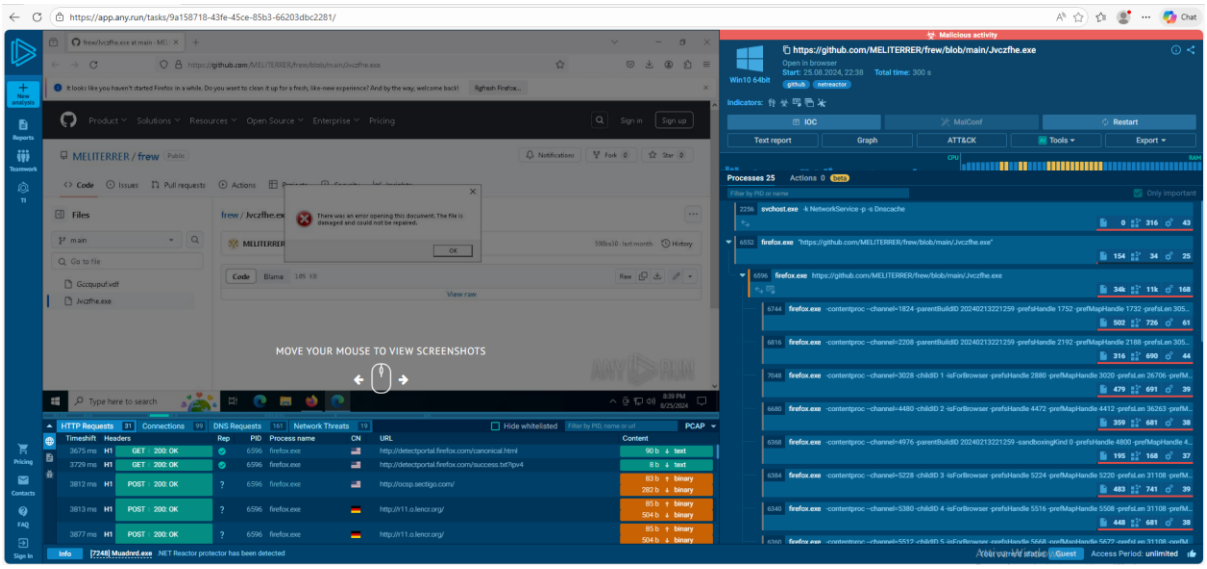
- **Hypervisor:** Oracle VM VirtualBox
- **Piattaforma di Analisi Dinamica (Sandbox):** anyrun
- **OS Target (nella Sandbox):** Windows
- **Browser web:** Per l'accesso e la navigazione del report di anyrun
- **Vettore analizzato:** Link GitHub contenente un eseguibile sospetto

Svolgimento dell'Analisi e Metodologia

Vettore d'Infezione e Tecniche di Ingegneria Sociale

Come primo passo ho osservato la schermata principale dell'esecuzione. L'analisi parte dal browser Firefox che naviga verso un URL specifico: https://github.com/MELITERRER/frew/blob/main/Jvczfhe.exe

A schermo compare un finto messaggio di errore: "There was an error opening this document. The file is damaged and could not be repaired." Questo è un'esca, il malware mostra un errore fittizio all'utente per fargli credere che il file semplicemente non funzioni, mentre in realtà il codice malevolo è già in esecuzione in background.



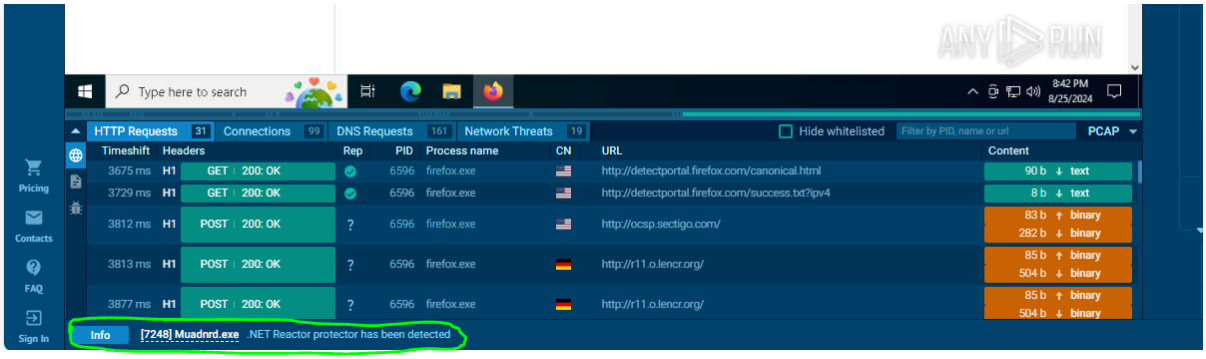
Navigare nell'Albero dei Processi (Process Tree)

A livello professionale l'albero dei processi è la prima cosa da guardare per capire chi ha generato cosa. Nel pannello a destra, sotto "Processes", posso vedere la catena di esecuzione. Al momento vedo molti processi firefox.exe, che è il comportamento standard del browser.

Cosa devo fare per ampliare la ricerca:

1. Guardando attentamente il pannello di destra ("Processes"). Scorro verso il basso. Cerco un processo che non sia Firefox. Spesso i processi malevoli sono evidenziati in rosso o arancione.

2. Dal pannello in basso a sinistra (barra blu), ho notato un'informazione cruciale: [Info] [7240] Muadnrd.exe .NET Reactor protector has been detected. Questo significa che un eseguibile chiamato Muadnd.exe è stato lanciato ed è protetto/offuscato con ".NET Reactor" per rendere difficile l'analisi ai ricercatori (Reverse Engineering).



Cos'è .NET Reactor e come contrastarlo?

Cos’è?

NET Reactor è uno strumento legittimo e commerciale sviluppato da Eziriz, utilizzato dagli sviluppatori software per proteggere la proprietà intellettuale delle loro applicazioni scritte in C# o VB.NET. A differenza dei programmi compilati in C o C++ che vengono tradotti in linguaggio macchina i programmi .NET vengono compilati in un linguaggio intermedio (CIL/MSIL). Questo li rende estremamente facili da "decompilare", permettendo a chiunque di leggere il codice sorgente originale quasi in chiaro. .NET Reactor previene questo problema applicando tecniche di offuscamento e compressione/incapsulamento.

Nel contesto della cybersecurity gli autori di malware (Threat Actors) abusano sistematicamente di packer e offuscatori legittimi come .NET Reactor per i seguenti motivi tattici:

1. **Evasione degli Antivirus (AV/EDR):** Modificando pesantemente la struttura del file eseguibile e criptandone il contenuto, l'hash del file cambia completamente e le firme (signatures) tradizionali degli antivirus non riescono a riconoscerlo come minaccia.
2. **Anti-Reverse Engineering:** .NET Reactor altera i nomi delle variabili, delle funzioni e delle classi rendendoli incomprensibili (es. sostituendoli con caratteri casuali), cripta le stringhe (nascondendo così gli indirizzi IP e gli URL dei server di Comando e Controllo) e implementa controlli Anti-Debugging.
3. **Ostacolo all'Analisi Statica:** Se provassi ad aprire Muadnd.exe con un decompilatore standard (come dnSpy o dotPeek) al di fuori della

sandbox, vedrei solo il codice "involucro" del packer, mentre il vero payload malevolo rimarrebbe nascosto e criptato, venendo decriptato in memoria solo durante l'esecuzione (da qui l'importanza vitale dell'analisi dinamica su ANY.RUN).

In sintesi, la presenza di .NET Reactor in un eseguibile scaricato in modo sospetto da GitHub è un fortissimo indicatore (Red Flag) della volontà dell'autore di nascondere intenzionalmente il comportamento del software per eludere le analisi dei gruppi di sicurezza.

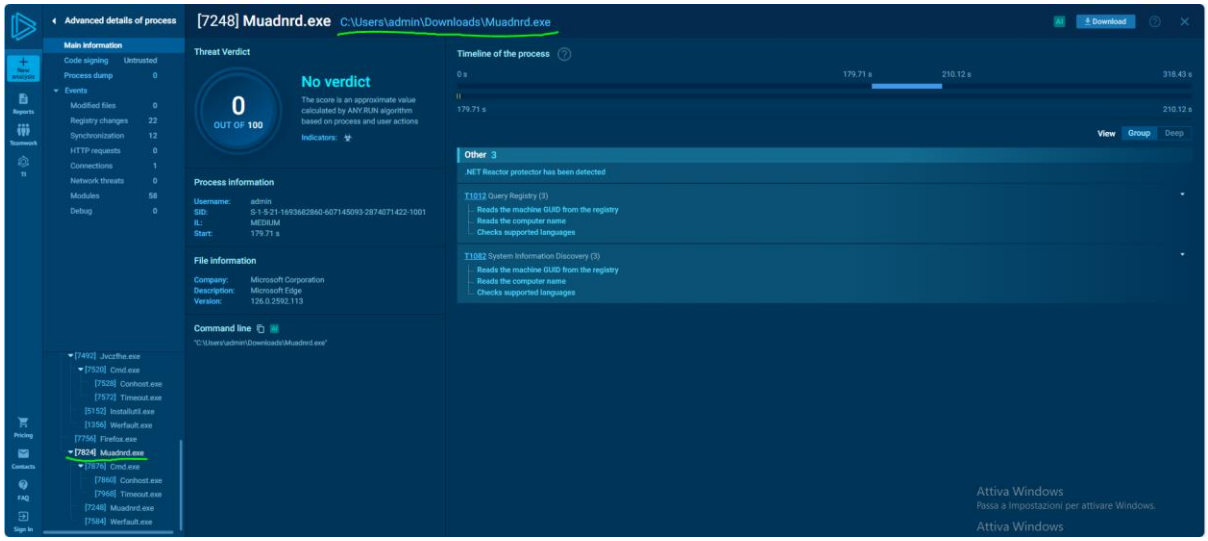
Come si può contrastare?

Dal punto di vista di un analista di sicurezza o di un membro del Blue Team, la scoperta di un file protetto da .NET Reactor richiede un approccio metodologico per "spacchettare" (unpacking) il file e rivelarne il vero codice sorgente. Per fare ciò, utilizzo due approcci principali:

1. **Unpacking Statico tramite Tool Dedicati:** Per offuscatori noti come .NET Reactor, la community di sicurezza ha sviluppato strumenti open-source specifici, come **de4dot**. Questo tool da riga di comando è in grado di riconoscere decine di offuscatori .NET e tentare di ripristinare automaticamente la struttura originale del file, rinominando le variabili con nomi leggibili e decriptando le stringhe. Se de4dot ha successo, posso poi aprire il file pulito su un decompilatore come dnSpy per un'analisi statica approfondita.
2. **Memory Dumping (Unpacking Dinamico):** Quando l'offuscamento è troppo complesso o customizzato, lascio che sia il malware stesso a fare il lavoro sporco. Poiché la CPU non può eseguire codice criptato, il malware deve forzatamente decriptare il suo payload (il codice malevolo vero e proprio) e scriverlo nella memoria RAM (spesso usando tecniche come il Process Hollowing). Sfruttando strumenti avanzati come Process Hacker o debugger come x64dbg, posso "congelare" il processo malevolo nel momento esatto in cui ha decriptato il payload e fare un "dump" (estrazione) di quella porzione di memoria, salvandola su disco come un nuovo file eseguibile finalmente in chiaro.

È proprio per evitare queste lunghe procedure manuali di unpacking statico che l'uso di una sandbox dinamica come anyrun risulta fondamentale: mi permette di osservare gli effetti del malware (connessioni, file creati) senza dover necessariamente decifrare il codice riga per riga.

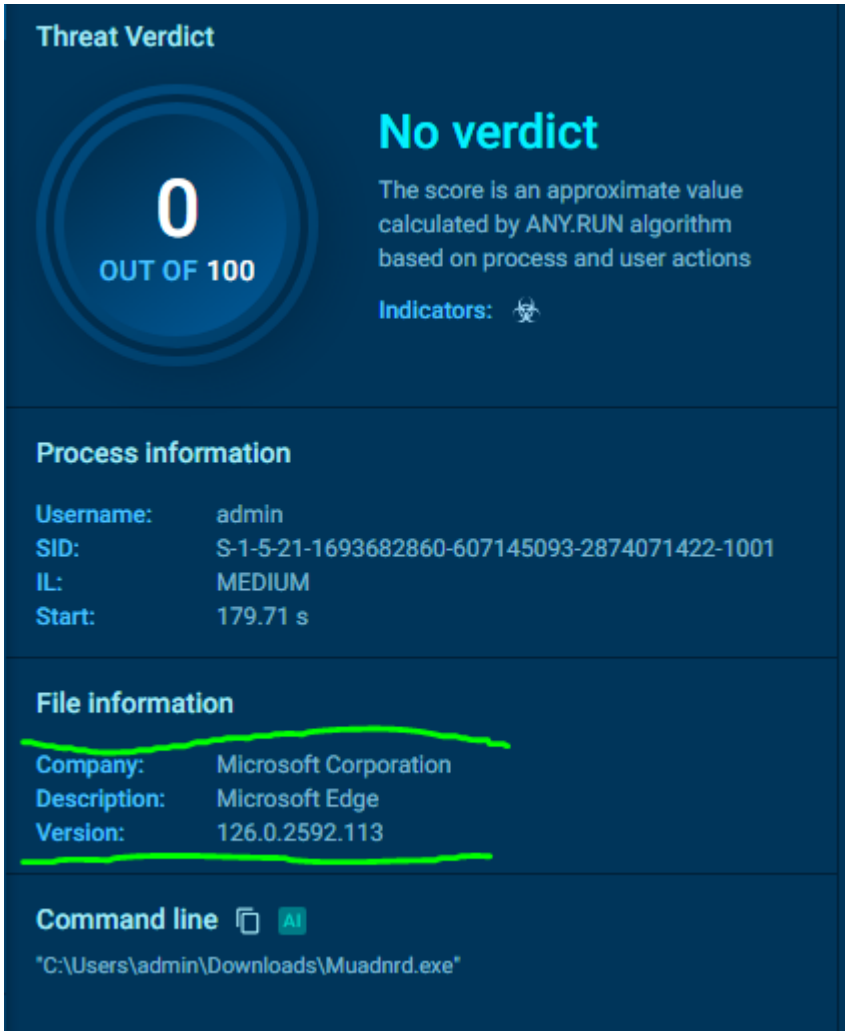
3. Clicco su quel processo sospetto (Muadnrd.exe) nell'albero di destra. Si aprirà una finestra chiamata "Process details" (Dettagli del processo). Da lì clicco sul pulsante "More Info".



cosa posso estrarre da questa schermata?

Navigando nell'interfaccia di anyrun, ho aperto i dettagli avanzati ("More Info") del processo Muadnrd.exe per esaminarne il comportamento a basso livello. Dalla dashboard di riepilogo ho potuto estrarre diversi Indicatori di Compromissione (IoC) comportamentali:

1. Tecniche di Mascheramento: Ispezionando la sezione "File information", ho notato una discrepanza evidente. L'eseguibile riporta come "Company" Microsoft Corporation e come "Description" Microsoft Edge. Questa è una classica tecnica di evasione: l'attaccante ha falsificato i metadati del payload malevolo per camuffarlo da componente di sistema legittimo, tentando di ingannare sia l'utente finale che eventuali controlli di sicurezza basati su whitelist superficiali.



2. Discovery e Profilazione dell'Ambiente: Il motore della sandbox ha mappato le azioni del file sulle tattiche MITRE **T1012 (Query Registry)** e **T1082 (System Information Discovery)**. Nello specifico, il malware interroga il registro di sistema per leggere il Machine GUID (l'identificativo univoco della macchina), il nome del computer e le lingue supportate. Questa è una chiara fase di profilazione: il malware sta verificando se si trova all'interno di un ambiente di analisi (sandbox) o se la lingua di sistema corrisponde ai suoi target prefissati. Se i parametri non lo soddisfano, il malware potrebbe terminare la sua esecuzione per non farsi analizzare.

3. Anomalie nella Catena di Esecuzione (Process Tree): Analizzando il pannello laterale sinistro contenente l'albero dei processi genitore/figlio, ho rilevato un comportamento tipico delle minacce avanzate legato al processo parallelo Jvczfhe.exe (PID 7492). Questo ha generato una sequenza sospetta di utility native di Windows:

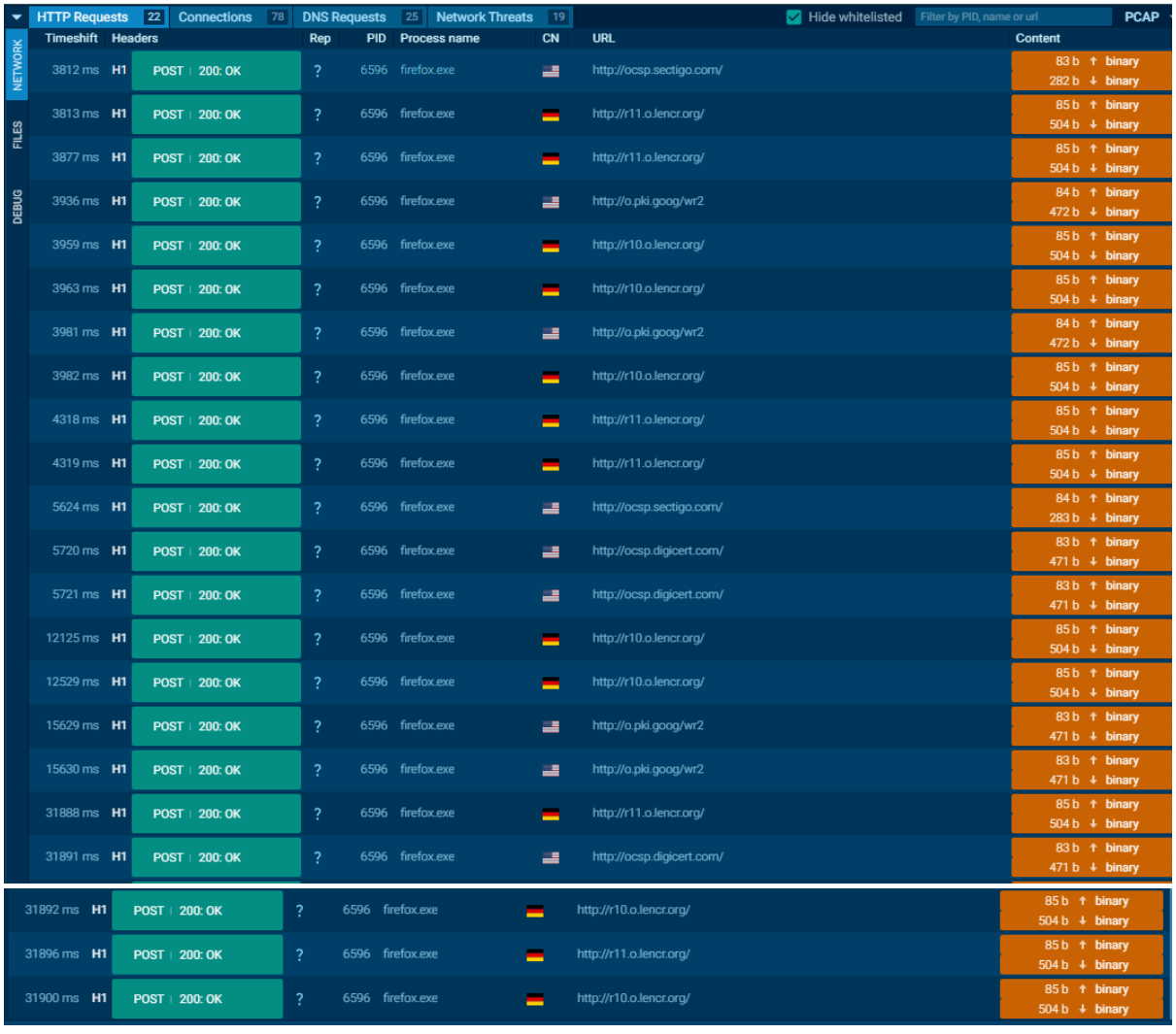
- **cmd.exe e timeout.exe**: Spesso eseguiti in sequenza per introdurre un ritardo forzato (evasion delay), sperando che la scansione dell'antivirus vada in timeout.
- **InstallUtil.exe**: Un binario legittimo del framework .NET. L'uso di questo strumento da parte del malware indica una tecnica Living off the Land (LOLBins), utilizzata per bypassare restrizioni come AppLocker ed eseguire codice non firmato.
- **WerFault.exe**: L'eseguibile per la segnalazione degli errori di Windows. La sua presenza in questa catena è altamente sospetta e spesso indicativa di un tentativo di Process Injection o Process Hollowing, dove il codice malevolo viene iniettato nello spazio di memoria di WerFault per nascondere le proprie tracce al Task Manager e all'EDR.

Analisi del Traffico di Rete (Network IoCs)

Un malware solitamente ha bisogno di comunicare con l'esterno per scaricare ulteriori moduli o per ricevere comandi (Server C2 - Command & Control).

Cosa devo fare:

- Guardo il pannello in basso a sinistra. Al momento sono sulla scheda **"HTTP Requests"**.



Per isolare le reali minacce dal traffico di sistema, ho attivato il filtro **"Hide whitelisted"**. Questa operazione ha ridotto il rumore di fondo, eliminando gran parte delle chiamate di telemetria standard di Windows e Firefox.

Nonostante il filtraggio rimangono ancora alcune richieste verso infrastrutture di certificazione (OCSP). Questo indica che il traffico

sospetto potrebbe essere incapsulato in sessioni crittografate che utilizzano questi certificati per apparire legittime. L'assenza di richieste HTTP esplicitamente malevole (come il download di ulteriori file .exe o script .ps1) sposta il focus dell'analisi verso i livelli più bassi della pila ISO/OSI: le connessioni TCP dirette e le interrogazioni DNS.

In un'ottica di difesa attiva, questo "silenzio" nel protocollo HTTP suggerisce che il payload malevolo stia utilizzando tecniche di comunicazione alternative per contattare il proprio server di Command & Control (C2), probabilmente per esfiltrare i dati raccolti durante la fase di profilazione del sistema vista in precedenza.

- Clicco sulla scheda **"Connections"**

HTTP Requests 22 | Connections 78 | DNS Requests 25 | Network Threats 19 | Hide whitelisted | Filter by PID, domain, name or ip | PCAP

NETWORK | FILES | DEBUG

Timeshift	Protocol	Rep	PID	Process name	CN	IP	Port	Domain	ASN	Traffic
BEFORE	TCP	?	1920	svchost.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	No Data
BEFORE	TCP	?	1048	RUXIMICS.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	No Data
BEFORE	TCP	?	2120	MoUsoCoreWorker.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	No Data
3517 ms	TCP	?	6596	firefox.exe		140.82.121.3	443	github.com	GITHUB	↑ 3 Kb ↓ 115 Kb
3732 ms	TCP	?	6596	firefox.exe		34.117.188.166	443	contile.services.mozilla.com	GOOGLE-CLOUD-PLATFORM	↑ 1 Kb ↓ 11 Kb
3735 ms	TCP	?	6596	firefox.exe		34.117.188.166	443	contile.services.mozilla.com	GOOGLE-CLOUD-PLATFORM	↑ 1 Kb ↓ 26 Kb
3739 ms	TCP	?	6596	firefox.exe		172.64.149.23	80	ocsp.sectigo.com	CLOUDFLARENET	↑ 851 b ↓ 2 Kb
3811 ms	TCP	?	6596	firefox.exe		184.24.77.69	80	r11.o.lencr.org	Akamai International B.V.	↑ 2 Kb ↓ 3 Kb
3828 ms	UDP	?	6596	firefox.exe		34.117.188.166	443	contile.services.mozilla.com	--	↑ 2 Kb ↓ 4 Kb
3829 ms	TCP	?	6596	firefox.exe		34.107.243.93	443	push.services.mozilla.com	GOOGLE	↑ 1001 b ↓ 4 Kb
3877 ms	TCP	?	6596	firefox.exe		184.24.77.74	80	r11.o.lencr.org	Akamai International B.V.	↑ 852 b ↓ 2 Kb
3942 ms	TCP	?	6596	firefox.exe		34.160.144.191	443	content-signature-2.cdn.mozil...	GOOGLE	↑ 4 Kb ↓ 105 Kb
3944 ms	UDP	?	6596	firefox.exe		34.117.188.166	443	contile.services.mozilla.com	--	↑ 2 Kb ↓ 4 Kb
3944 ms	UDP	?	6596	firefox.exe		34.107.243.93	443	push.services.mozilla.com	--	↑ 2 Kb ↓ 4 Kb
3946 ms	TCP	?	6596	firefox.exe		184.24.77.81	80	r10.o.lencr.org	Akamai International B.V.	↑ 1 Kb ↓ 3 Kb
3959 ms	TCP	?	6596	firefox.exe		184.24.77.81	80	r10.o.lencr.org	Akamai International B.V.	↑ 852 b ↓ 2 Kb
3963 ms	TCP	?	6596	firefox.exe		34.149.100.209	443	prod.remote-settings.prod.we...	GOOGLE	↑ 5 Kb ↓ 619 Kb
3970 ms	TCP	?	6596	firefox.exe		34.107.243.93	443	push.services.mozilla.com	GOOGLE	↑ 2 Kb ↓ 1 Kb
4054 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	↑ 15 Kb ↓ 1 Mb
4124 ms	TCP	?	6596	firefox.exe		185.199.108.133	443	avatars.githubusercontent.com	FASTLY	↑ 1 Kb ↓ 6 Kb
4153 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	No Data
4162 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	No Data
4164 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	No Data
4168 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	No Data
4171 ms	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	No Data
4309 ms	TCP	?	6596	firefox.exe		34.36.165.17	443	tiles-cdn.prod.ads.prod.webse...	GOOGLE-CLOUD-PLATFORM	↑ 969 b ↓ 4 Kb
4317 ms	TCP	?	6596	firefox.exe		34.36.165.17	443	tiles-cdn.prod.ads.prod.webse...	GOOGLE-CLOUD-PLATFORM	↑ 1 Kb ↓ 29 Kb
5557 ms	TCP	?	6596	firefox.exe		140.82.114.21	443	collector.github.com	GITHUB	↑ 7 Kb ↓ 7 Kb
5576 ms	TCP	?	6596	firefox.exe		140.82.114.21	443	collector.github.com	GITHUB	↑ 943 b ↓ 4 Kb
5617 ms	TCP	?	6596	firefox.exe		140.82.114.21	443	collector.github.com	GITHUB	↑ 943 b ↓ 4 Kb
5621 ms	TCP	?	6596	firefox.exe		140.82.121.6	443	api.github.com	GITHUB	↑ 3 Kb ↓ 7 Kb
5623 ms	TCP	?	6596	firefox.exe		140.82.114.21	443	collector.github.com	GITHUB	No Data
12123 ms	TCP	?	6596	firefox.exe		54.71.162.254	443	shavar.services.mozilla.com	AMAZON-02	↑ 2 Kb ↓ 4 Kb
12125 ms	TCP	?	6596	firefox.exe		184.24.77.81	80	r10.o.lencr.org	Akamai International B.V.	↑ 852 b ↓ 2 Kb
12529 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 1 Kb ↓ 8 Kb

HTTP Requests 22 | Connections 78 | DNS Requests 25 | Network Threats 19 | Hide whitelisted | Filter by PID, domain, name or ip | PCAP

NETWORK | FILES | DEBUG

Timeshift	Protocol	Rep	PID	Process name	CN	IP	Port	Domain	ASN	Traffic
12532 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 10 Kb
12534 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 1 Mb
13324 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 297 Kb
13525 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 3 Kb
13527 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 5 Kb
14629 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 1 Kb
14632 ms	TCP	?	6596	firefox.exe		34.120.158.37	443	tracking-protection.cdn.mozill...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 4 Kb
14634 ms	TCP	?	6596	firefox.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	↑ 1 Kb ↓ 111 Kb
19729 ms	TCP	?	2120	MoUsoCoreWorker.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 860 b ↓ 4 Kb
19731 ms	TCP	?	2268	svchost.exe		20.190.159.0	443	login.live.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 5 Kb ↓ 14 Kb
19734 ms	TCP	?	2120	MoUsoCoreWorker.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 1 Kb ↓ 18 Kb
20729 ms	TCP	?	2120	MoUsoCoreWorker.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 2 Kb ↓ 9 Kb
20731 ms	TCP	?	2120	MoUsoCoreWorker.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 1 Kb ↓ 4 Kb
27937 ms	TCP	?	7816	SIHClient.exe		40.127.169.103	443	slscr.update.microsoft.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 752 b ↓ 3 Kb
28940 ms	TCP	?	7816	SIHClient.exe		20.166.126.56	443	fe3cr.delivery.mp.microsoft.co...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 602 b ↓ 3 Kb
28942 ms	TCP	?	7816	SIHClient.exe		40.127.169.103	443	slscr.update.microsoft.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 584 b ↓ 3 Kb
28947 ms	TCP	?	7816	SIHClient.exe		40.127.169.103	443	slscr.update.microsoft.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 750 b ↓ 3 Kb
31859 ms	TCP	?	6596	firefox.exe		34.149.100.209	443	prod.remote-settings.prod.we...	GOOGLE	↑ 536 b ↓ 3 Kb
31884 ms	TCP	?	6596	firefox.exe		35.201.103.21	443	normandy.cdn.mozilla.net	GOOGLE	↑ 1 Kb ↓ 5 Kb
31887 ms	TCP	?	6596	firefox.exe		35.244.181.201	443	star-mini.c10r.facebook.com	GOOGLE	↑ 1 Kb ↓ 5 Kb
31890 ms	TCP	?	6596	firefox.exe		35.190.72.216	443	prod.classify-client.prod.webs...	GOOGLE	↑ 1 Kb ↓ 4 Kb
31895 ms	TCP	?	6596	firefox.exe		34.98.75.36	443	classify-client.services.mozilla...	GOOGLE	↑ 1 Kb ↓ 4 Kb
31900 ms	TCP	?	6596	firefox.exe		34.117.121.53	443	attachments.prod.remote-sett...	GOOGLE-CLOUD-PLATFORM	↑ 904 b ↓ 844 Kb
31946 ms	UDP	?	6596	firefox.exe		35.190.72.216	443	prod.classify-client.prod.webs...	--	↑ 2 Kb ↓ 4 Kb
31947 ms	TCP	?	6596	firefox.exe		34.160.144.191	443	content-signature-2.cdn.mozil...	GOOGLE	↑ 8 Kb ↓ 226 Kb
39652 ms	TCP	?	7492	Jvczfhe.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	↑ 417 b ↓ 5 Mb
44558 ms	TCP	?	2268	svchost.exe		20.190.159.0	443	login.live.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 5 Kb ↓ 14 Kb
44563 ms	TCP	?	1920	svchost.exe		40.127.240.158	443	settings-win.data.microsoft.c...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 2 Kb ↓ 4 Kb
51472 ms	TCP	?	6596	firefox.exe		34.160.144.191	443	content-signature-2.cdn.mozil...	GOOGLE	↑ 983 b ↓ 9 Kb
51498 ms	TCP	?	6596	firefox.exe		23.53.40.162	80	ciscobinary.openh264.org	Akamai International B.V.	↑ 305 b ↓ 480 Kb
55663 ms	TCP	?	5152	InstallUtil.exe		91.92.253.47	7702	egehgdehjbhjtre.duckdns.org	--	No Data
56761 ms	TCP	?	2268	svchost.exe		20.190.159.0	443	login.live.com	MICROSOFT-CORP-MSN-AS-BLO...	↑ 5 Kb ↓ 15 Kb
56763 ms	TCP	?	1356	WerFault.exe		104.208.16.94	443	watson.events.data.microsoft...	MICROSOFT-CORP-MSN-AS-BLO...	↑ 6 Kb ↓ 5 Kb
100.80 s	TCP	?	6596	firefox.exe		34.36.165.17	443	tiles-cdn.prod.ads.prod.webse...	GOOGLE-CLOUD-PLATFORM	↑ 2 Kb ↓ 29 Kb
102.70 s	TCP	?	6596	firefox.exe		140.82.121.3	443	github.com	GITHUB	↑ 3 Kb ↓ 115 Kb

102.71 s	TCP	?	6596	firefox.exe		140.82.112.21	443	glb-db52c2cf8be544.github.c...	GITHUB	10 Kb	8 Kb
102.71 s	TCP	?	6596	firefox.exe		185.199.109.154	443	github.githubassets.com	FASTLY	15 Kb	1 Mb
102.71 s	TCP	?	6596	firefox.exe		185.199.108.133	443	avatars.githubusercontent.com	FASTLY	1 Kb	8 Kb
104.40 s	TCP	?	6596	firefox.exe		140.82.121.6	443	api.github.com	GITHUB	3 Kb	6 Kb
105.40 s	TCP	?	6596	firefox.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	1 Kb	112 Kb
119.70 s	TCP	?	2268	svchost.exe		20.190.159.0	443	login.live.com	MICROSOFT-CORP-MSN-AS-BLO...	5 Kb	15 Kb
128.91 s	TCP	?	7824	Muadnrd.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	415 b	5 Mb
177.06 s	TCP	?	7584	WerFault.exe		20.42.65.92	443	watson.events.data.microsoft...	MICROSOFT-CORP-MSN-AS-BLO...	6 Kb	5 Kb

L'attivazione del filtro "Hide whitelisted" ha permesso di isolare le comunicazioni non correlate ai servizi standard di Windows o del browser. Dall'analisi della tabella delle connessioni (Connections), emerge un Indicatore di Compromissione (IoC) di importanza critica:

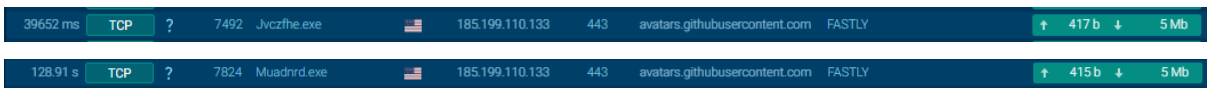
1. Individuazione del Server di Comando e Controllo (C2): Il dettaglio più allarmante si trova nella riga relativa al processo **InstallUtil.exe (PID 5152)**. Questo processo, che abbiamo precedentemente identificato come un LOLBin abusato dal malware, ha stabilito una connessione di rete verso:

55663 ms	TCP	?	5152	InstallUtil.exe		91.92.253.47	7702	egehgdehjbhjtre.duckdns.org	–	No Data	

- **Dominio:** egehgdehjbhjtre.duckdns.org
- **IP:** 91.92.253.47
- **Porta:** 7702
- **Paese (CN):** Bulgaria (BG)

L'uso di **DuckDNS** (un servizio di DNS dinamico gratuito) è una tecnica estremamente comune tra gli sviluppatori di malware per creare infrastrutture di comando flessibili e difficili da abbattere. La porta **7702** non è una porta standard (come la 80 o la 443) il che conferma che non si tratta di normale traffico web, ma di un canale di comunicazione diretto tra il malware e l'attaccante.

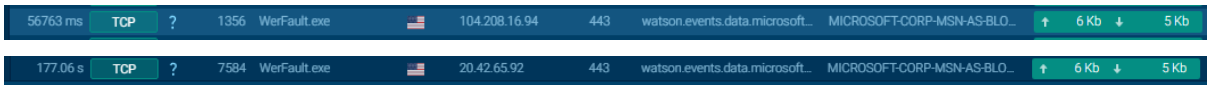
2. Attività dei Processi Offuscati: Sempre nella tabella Connections, osserviamo l'attività dei processi figli che abbiamo analizzato nei "More Info":



39652 ms	TCP	?	7492	Jvczfhe.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	417 b	5 Mb
128.91 s	TCP	?	7824	Muadnrd.exe		185.199.110.133	443	avatars.githubusercontent.com	FASTLY	415 b	5 Mb

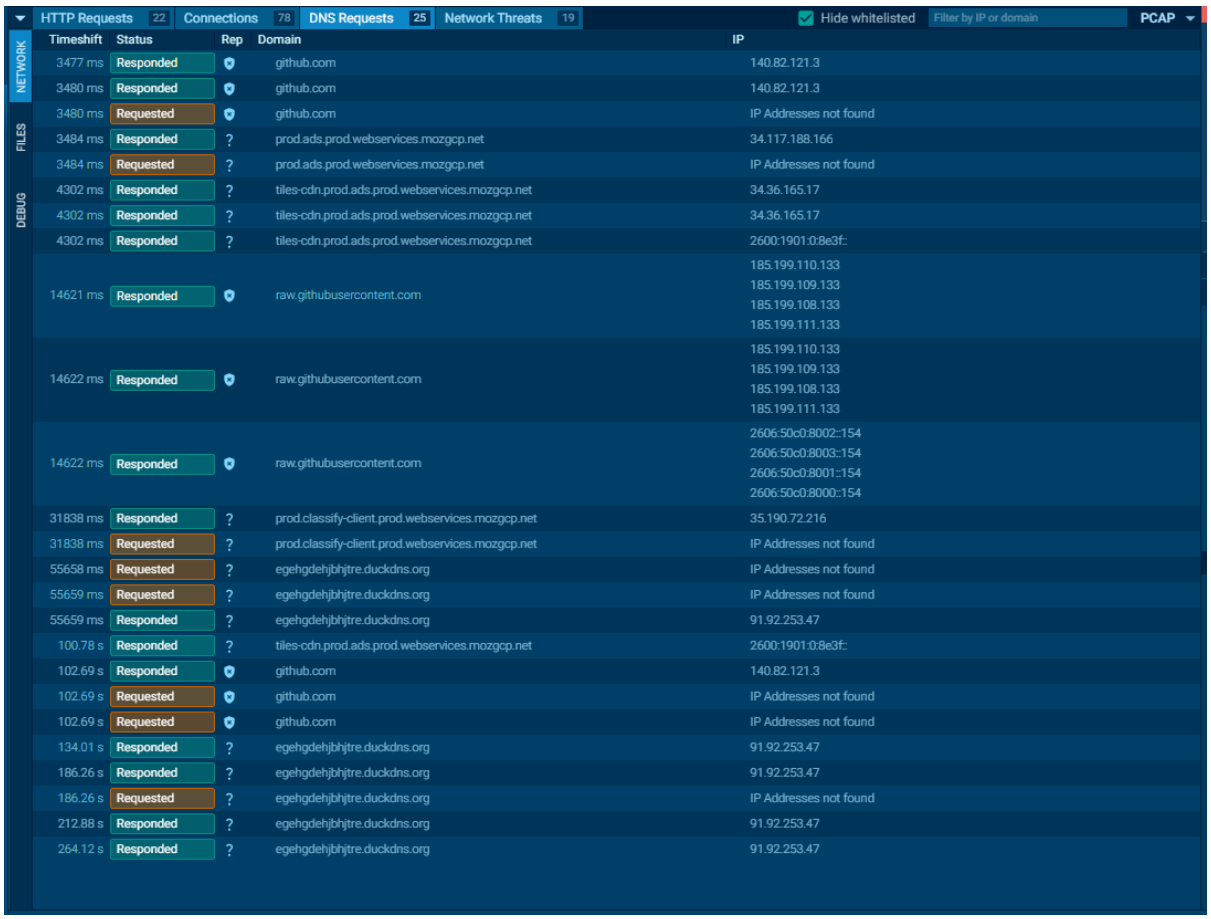
Jvczfhe.exe (PID 7492) e **Muadnrd.exe (PID 7824)** effettuano connessioni verso gli asset di GitHub (avatars.githubusercontent.com) tramite l'infrastruttura FASTLY. Questo indica che il malware continua a scaricare

componenti o configurazioni mascherandosi da traffico verso siti affidabili (un'estensione della tecnica di Mascheramento).



WerFault.exe (PID 1356) contatta i server di telemetria Microsoft. Come ipotizzato nella fase precedente l'attaccante sta probabilmente sfruttando questo processo legittimo per inviare dati rubati o segnalazioni di stato, sperando che il traffico passi inosservato poiché diretto a domini Microsoft.

- Clicco sulla scheda **"DNS Requests"**.



L'analisi dei log DNS filtrati conferma la catena di risoluzione dei nomi utilizzata dall'attaccante per l'invio dei dati e il mantenimento della persistenza. Da 161 a 25 record rimasti:

1. Risoluzione del Vettore di Infezione (GitHub): Si osservano le query per github.com e raw.githubusercontent.com, risolte sugli indirizzi IP 140.82.121.3 e la serie 185.199.110.133. Questo conferma che il malware ha utilizzato le API di GitHub per scaricare il payload iniziale o configurazioni aggiuntive durante l'esecuzione. L'uso di domini ad alta reputazione permette di evadere i filtri basati su blacklist DNS.

2. Risoluzione del Command & Control (C2) - DuckDNS: L'elemento più critico è la risoluzione ripetuta del dominio **egehgdehjbhjtre.duckdns.org**.

- Il malware interroga il server DNS per ottenere l'indirizzo IP associato a questo sottodominio dinamico. La sandbox ha registrato la risposta con l'IP **91.92.253.47**.
- **Analisi temporale:** Le richieste iniziano intorno ai 55 secondi (55658 ms) e continuano fino alla fine dell'analisi (264.12 s), indicando un tentativo persistente di "beaconing" (richieste a intervalli regolari) verso l'attaccante.

3. Servizi di Supporto e Analisi: Si notano risoluzioni per prod.classify-client.prod.webservices.mozgcp.net e altri servizi di telemetria legati a Google Cloud e Mozilla. Anche se filtrati parzialmente, rimangono visibili perché strettamente correlati all'attività di navigazione effettuata durante l'infezione.

- Le connessioni verso domini strani o indirizzi IP non associati a servizi noti saranno i nostri **IoC di rete**.

Ora che ho tutti i dati necessari posso creare la tabella finale del mio report:

Tipo IoC	Valore	Descrizione
Dominio C2	`egehgdehjbhjtre.duckdns.org`	Server di comando e controllo (Bulgaria)
Indirizzo IP	`91.92.253.47`	Server remoto collegato a InstallUtil.exe
Porta	`7702`	Canale di comunicazione non standard
Processi	`Muadnrd.exe`, `Jvczfhe.exe`, `InstallUtil.exe`	Eseguibili malevoli o abusati

Conclusione

L'analisi dinamica ha confermato che l'evento in esame non è un falso positivo, ma un'infezione attiva da Remote Access Trojan (RAT). La minaccia ha mostrato un elevato grado di sofisticazione, utilizzando tecniche di Mascheramento (camuffandosi da Microsoft Edge) e sfruttando binari di sistema legittimi come InstallUtil.exe per eseguire codice malevolo in memoria (tecnica Living off the Land).

Il vettore d'attacco iniziale è stato identificato nel download di un payload ospitato su GitHub, un dominio ad alta reputazione scelto appositamente per eludere i controlli di sicurezza. La fase di post-infezione ha rivelato un'attività di Command & Control (C2) persistente verso il dominio dinamico egehgdehjbhjtre.duckdns.org (IP 91.92.253.47) sulla porta non standard 7702.

In sintesi, la combinazione di tecniche di evasione, profilazione del sistema e l'instaurazione di un canale di comunicazione diretto verso l'estero confermano la compromissione totale dell'host.