# Build Week 3 – Traccia Extra 1

## Analisi del codice sorgente di Mydoom in ambiente FlareVM

---

# Executive Summary

Nel presente elaborato è stata condotta un'**analisi statica del codice sorgente del malware Mydoom, utilizzando un ambiente controllato basato su FlareVM**.
L'obiettivo dell'attività è stato **comprendere la struttura interna del malware, identificarne le principali funzionalità operative e valutarne i meccanismi di propagazione, persistenza ed eventuale comunicazione con infrastrutture esterne.**

L'analisi ha evidenziato la **presenza di componenti dedicate alla propagazione tramite protocollo SMTP, all'apertura di connessioni di rete mediante API Winsock e a tecniche di persistenza attraverso modifiche al registro di sistema Windows**. Sono inoltre emersi elementi riconducibili a comportamenti tipici dei worm di prima generazione, con funzionalità orientate alla diffusione massiva e alla possibile esecuzione di attività di tipo DDoS.

L'attività dimostra come lo studio del codice sorgente consenta di ricostruire la logica operativa di un malware e di individuare indicatori utili alla definizione di strategie di difesa e mitigazione.

# Introduzione

L'analisi del codice sorgente rappresenta una delle metodologie più approfondite nell'ambito della malware analysis, in quanto consente di esaminare direttamente la logica implementativa del software malevolo, senza dover necessariamente eseguirne il payload.

**E' stato analizzato il sorgente del worm Mydoom in ambiente isolato tramite FlareVM**, distribuzione Windows progettata per attività di reverse engineering e threat analysis.
L'obiettivo dell'esercitazione è stato individuare:

- Le modalità di propagazione del malware
- I meccanismi di persistenza nel sistema operativo
- Le funzionalità di comunicazione di rete (eventuale C2)
- Possibili tecniche di evasione o anti-analysis

L’analisi è stata condotta tramite strumenti integrati in FlareVM, tra cui PowerShell per l’ispezione del codice, ricerca di pattern sospetti e calcolo hash per garantire tracciabilità forense.

# Preparazione Ambiente (FlareVM)

### Ambiente utilizzato

- VM Windows 10 con **FlareVM**
- Strumenti principali:
    - PowerShell
    - 7-Zip
    - Notepad++
    - Cyberchef per decodificare rot13

---

# Creazione Cartella di Lavoro

Aprire PowerShell come Administrator:

**mkdir C:\BW3**
**cd C:\BW3**
**mkdir Mydoom**
**cd Mydoom**

Verificare percorso:

**pwd**

---

```
FLARE-VM 02/25/2026 10:00:10
PS C:\Windows\system32 > mkdir C:\BW3
>> cd C:\BW3
>> mkdir Mydoom
>> cd Mydoom
>> pwd


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/25/2026  10:01 AM                BW3


    Directory: C:\BW3


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/25/2026  10:01 AM                Mydoom

Drive        : C
Provider     : Microsoft.PowerShell.Core\FileSystem
ProviderPath : C:\BW3\Mydoom
Path         : C:\BW3\Mydoom
```

PowerShell con percorso:

**C:\BW3\Mydoom**

---

# Download Archivio Sorgente

Scaricare l'archivio Mydoom dalla repository indicata nel progetto.

In PowerShell:

**Invoke-WebRequest -Uri "https://github.com/akir4d/MalwareSourceCode/raw/main/Win32/Win32.Mydoom.a.7z" -OutFile "Win32.Mydoom.a.7z"**

Verificare presenza file:

**dir**

---

```
FLARE-VM 02/25/2026 10:01:36
PS C:\BW3\Mydoom > Invoke-WebRequest -Uri "https://github.com/akir4d/MalwareSourceCode/raw/main/Win32/Win32.Mydoom.a.7z"
 -OutFile "Win32.Mydoom.a.7z"
FLARE-VM 02/25/2026 10:05:23
PS C:\BW3\Mydoom > dir


    Directory: C:\BW3\Mydoom


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/25/2026  10:05 AM          27019 Win32.Mydoom.a.7z


FLARE-VM 02/25/2026 10:05:33
PS C:\BW3\Mydoom >
```

- Download completato

- Output `dir` con il file `.7z` visibile

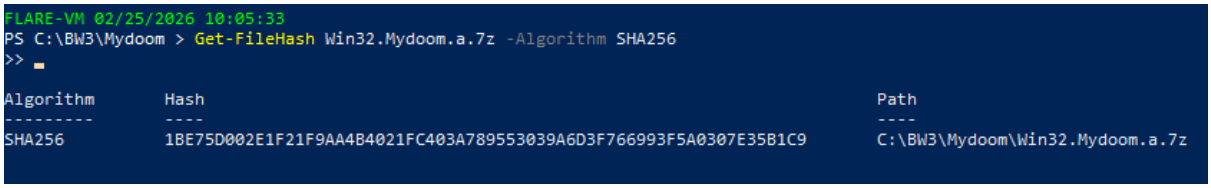
---

# Calcolo Hash (Tracciabilità Forense)

In FlareVM:

**Get-FileHash Win32.Mydoom.a.7z -Algorithm SHA256**

---




Output con SHA256 completo visibile.

---

# Estrazione Archivio

Metodo PowerShell:

**7z x Win32.Mydoom.a.7z -oC:\BW3\Mydoom\extracted**

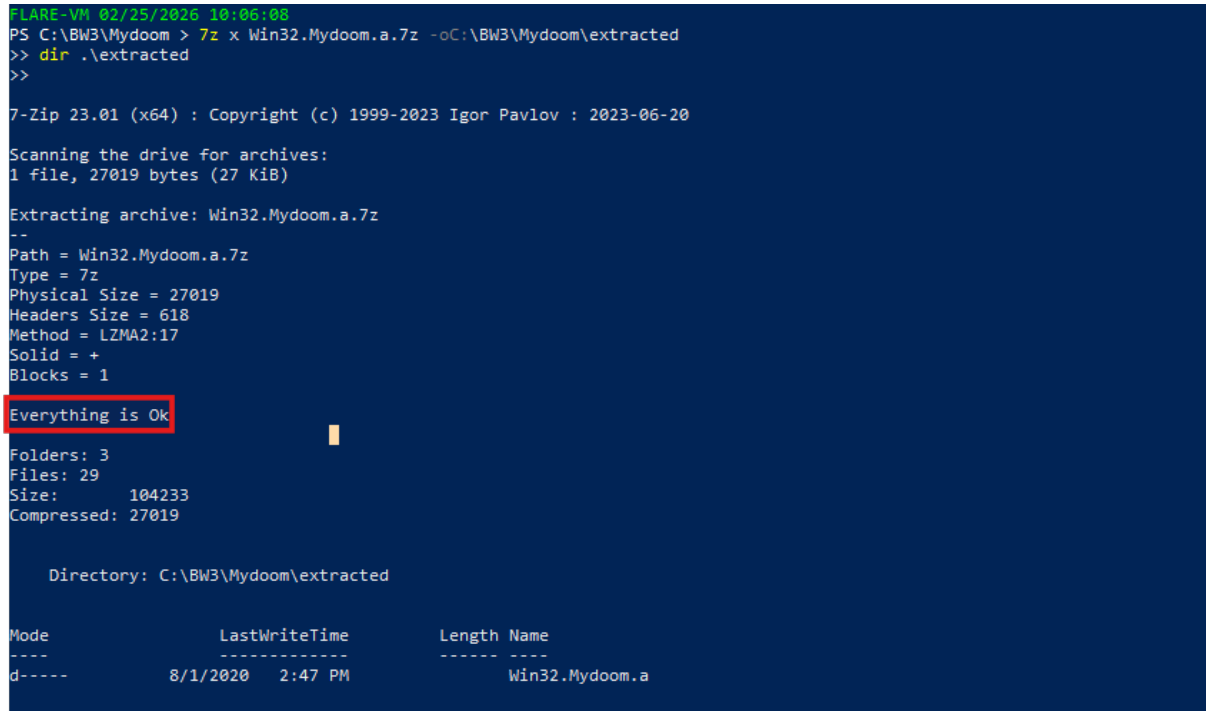
Oppure tramite GUI:

- Tasto destro → 7-Zip → Extract Here

Verificare contenuto:

**dir .\extracted**

---




- Output estrazione “Everything is Ok”
- Cartella extracted con file sorgenti visibili

---

# Analisi Strutturale del Codice

## Identificazione File Rilevanti

Ricercare estensioni tipiche:

**dir .\extracted -Recurse -Include *.c,*.cpp,*.h,*.asm**

---

```
FLARE-VM 02/25/2026 10:10:42
PS C:\BW3\Mydoom > dir .\extracted -Recurse -Include *.c,*.cpp,*.h,*.asm
>>


    Directory: C:\BW3\Mydoom\extracted\Win32.Mydoom.a\work


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/1/2020   2:47 PM            716 bin2c.c
-a----          8/1/2020   2:47 PM           1232 cleanpe.cpp
-a----          8/1/2020   2:47 PM            462 crypt1.c
-a----          8/1/2020   2:47 PM            416 rot13.c


    Directory: C:\BW3\Mydoom\extracted\Win32.Mydoom.a\xproxy


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/1/2020   2:47 PM           2349 client.c
-a----          8/1/2020   2:47 PM           9945 xproxy.c


    Directory: C:\BW3\Mydoom\extracted\Win32.Mydoom.a


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          8/1/2020   2:47 PM           7177 lib.c
-a----          8/1/2020   2:47 PM            658 lib.h
-a----          8/1/2020   2:47 PM           7732 main.c
-a----          8/1/2020   2:47 PM          14645 massmail.c
-a----          8/1/2020   2:47 PM            600 massmail.h
-a----          8/1/2020   2:47 PM          11553 msg.c
-a----          8/1/2020   2:47 PM             90 msg.h
-a----          8/1/2020   2:47 PM           1622 p2p.c
-a----          8/1/2020   2:47 PM          11576 scan.c
-a----          8/1/2020   2:47 PM            138 scan.h
-a----          8/1/2020   2:47 PM           2398 sco.c
-a----          8/1/2020   2:47 PM             81 sco.h
-a----          8/1/2020   2:47 PM          10463 xdns.c
-a----          8/1/2020   2:47 PM            225 xdns.h
-a----          8/1/2020   2:47 PM           8718 xsmtp.c
-a----          8/1/2020   2:47 PM            121 xsmtp.h
-a----          8/1/2020   2:47 PM           9095 zipstore.c
-a----          8/1/2020   2:47 PM            119 zipstore.h
```

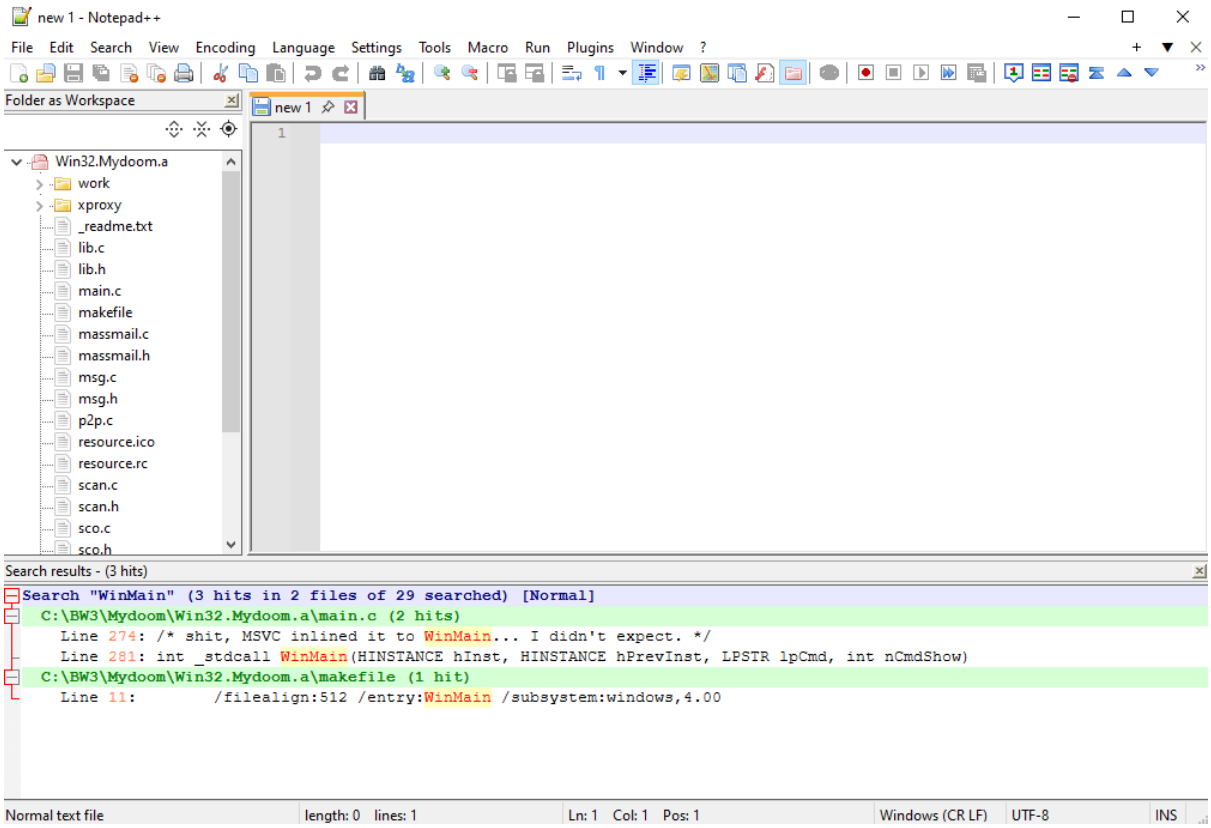
Lista file C/C++ visibile

---

# Analisi Tecnica del Codice (Core Malware Analysis)

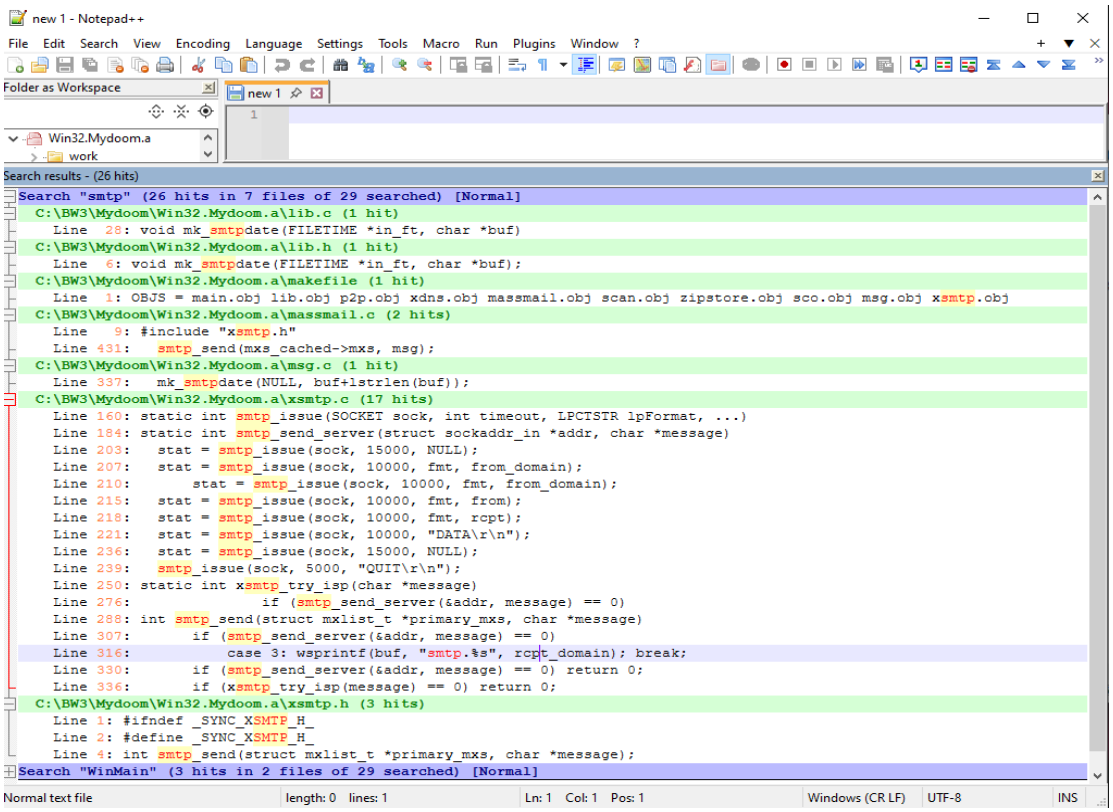
Aprire file principali con:

- VS Code
- Notepad++




# Propagazione

Cercare riferimenti SMTP:



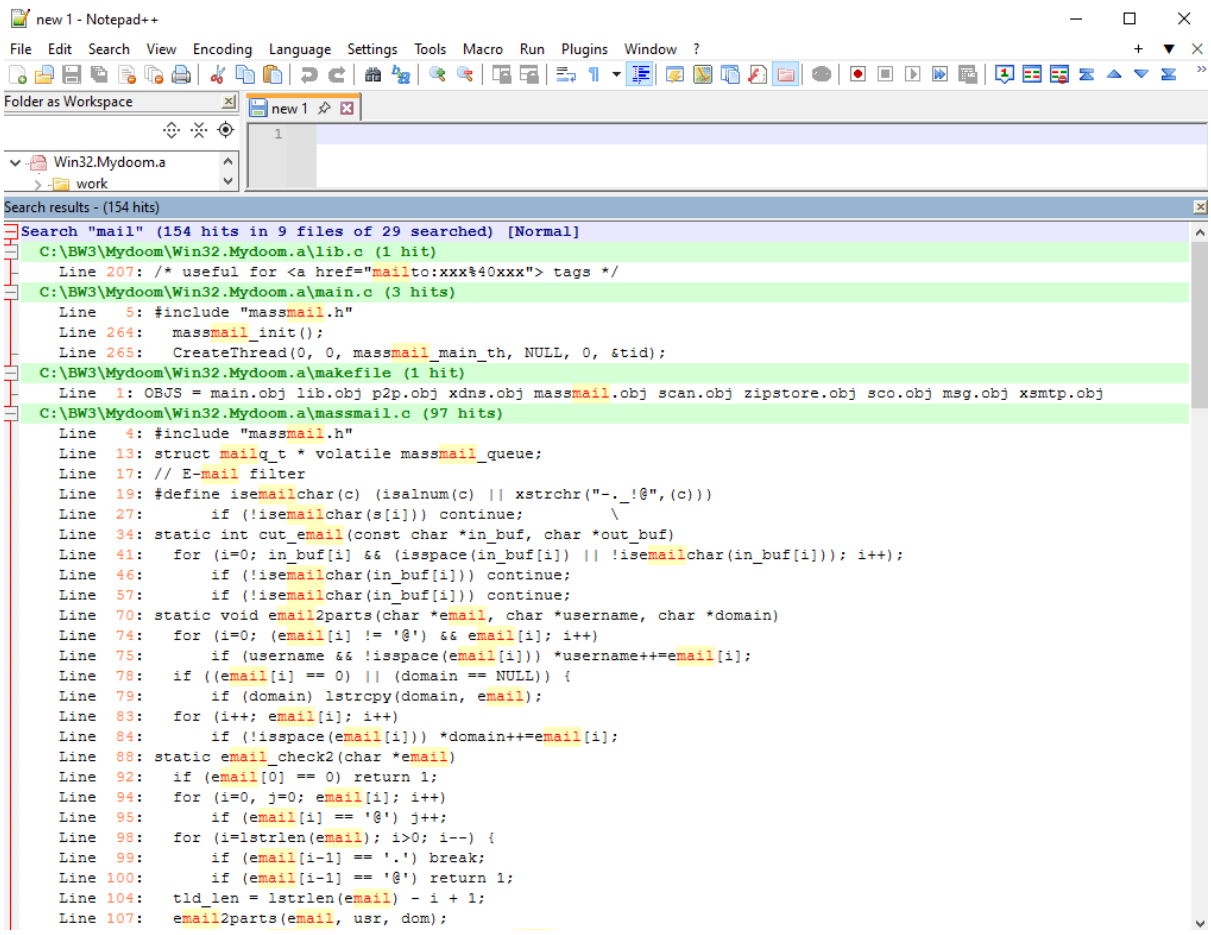
## Analisi della Propagazione SMTP

La ricerca della stringa “**smtp**” nel codice sorgente **ha evidenziato la presenza di un modulo completo dedicato all’invio di email tramite protocollo SMTP**. In particolare, i file `massmail.c` e `xsmtp.c` implementano rispettivamente la logica di mass-mailing e la gestione diretta dei comandi SMTP (MAIL FROM, RCPT TO, DATA, QUIT) tramite socket TCP.

Il file `msg.c` si occupa della **costruzione del contenuto del messaggio**, mentre la funzione `mk_smtpdate()` **genera correttamente l’header Date secondo lo standard SMTP.**

Questi elementi confermano che Mydoom integra un motore SMTP interno e si propaga autonomamente inviando email con allegato malevolo, senza dipendere da client di posta installati sul sistema.

**Cercare riferimenti mail:**

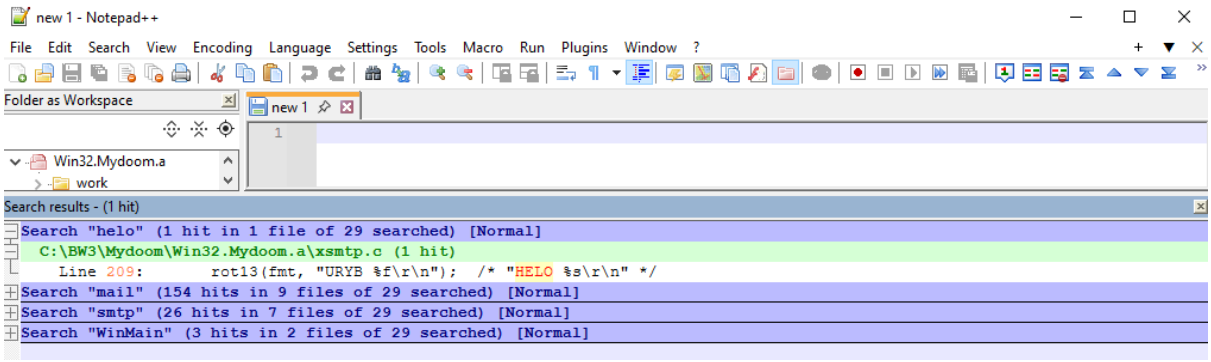



**Search mail:** La ricerca della stringa “**mail**” **evidenzia la presenza di un modulo strutturato per la gestione degli indirizzi email**. Il file `massmail.c` **implementa funzioni**

**di parsing, validazione e suddivisione degli indirizzi**, mentre `main.c` **avvia un thread dedicato all'attività di mass-mailing.**

Questi elementi confermano che Mydoom automatizza l'individuazione e l'utilizzo di indirizzi email per la propria propagazione, rafforzando la sua natura di worm a diffusione tramite posta elettronica.

**Cercare riferimenti: helo**

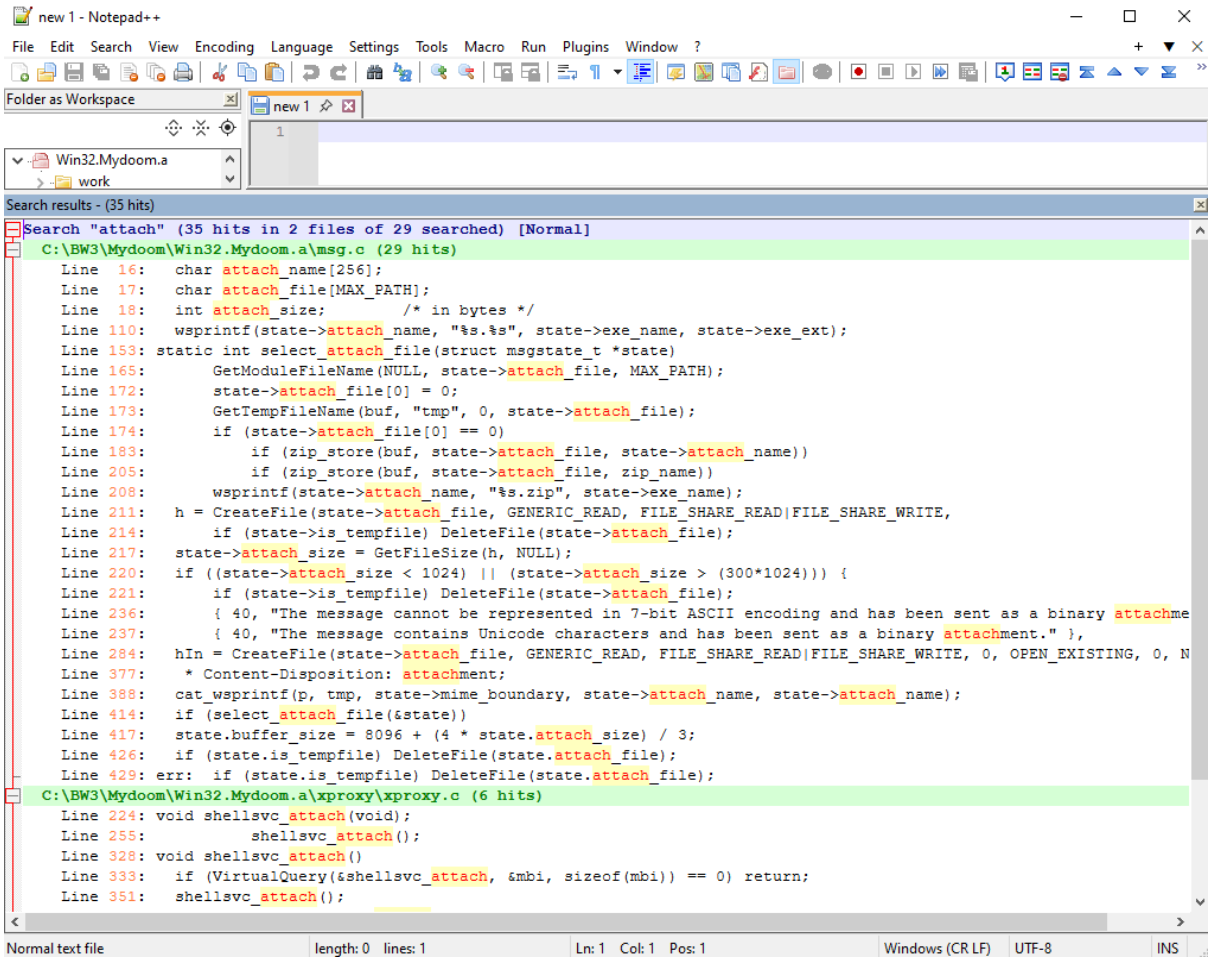



**Analisi della ricerca helo:** La presenza del comando **HELO** all'interno del modulo `xsmtp.c` **conferma l'implementazione diretta del protocollo SMTP nel codice del malware**. Questo rappresenta un'ulteriore evidenza della natura worm di Mydoom, **progettato per instaurare autonomamente connessioni SMTP e propagarsi tramite invio massivo di email malevole.**

**Cercare riferimenti rcpt:**

**Analisi della ricerca rcpt:** La presenza del comando **RCPT** TO nel modulo `xsmtp.c` **rappresenta un'ulteriore prova dell'implementazione completa del protocollo SMTP all'interno del malware**. Questo conferma che Mydoom è progettato per gestire autonomamente la selezione dei destinatari e l'invio massivo di email, consolidando la sua natura di worm a propagazione tramite posta elettronica.

**Cercare riferimenti attach:**




**Analisi della Ricerca “attach”:** L’analisi della stringa “**attach**” dimostra che Mydoom integra un modulo per la generazione e gestione dinamica di allegati malevoli, includendo compressione, codifica e inserimento nel messaggio SMTP. Questo **conferma che il malware non solo invia email, ma allega automaticamente il proprio payload eseguibile per garantire la propagazione del worm.**

Visionando il codice e isolandolo, è stato scoperto che **rot13 è un algoritmo di codifica per mascherare le funzioni malevoli del programma**, implementa il classico algoritmo a sostituzione monoalfabetica rot13 (cifrario).

Rot13 è utilizzato per mascherare le stringhe critiche di codice, ad esempio chiavi di registro e indirizzi IP.

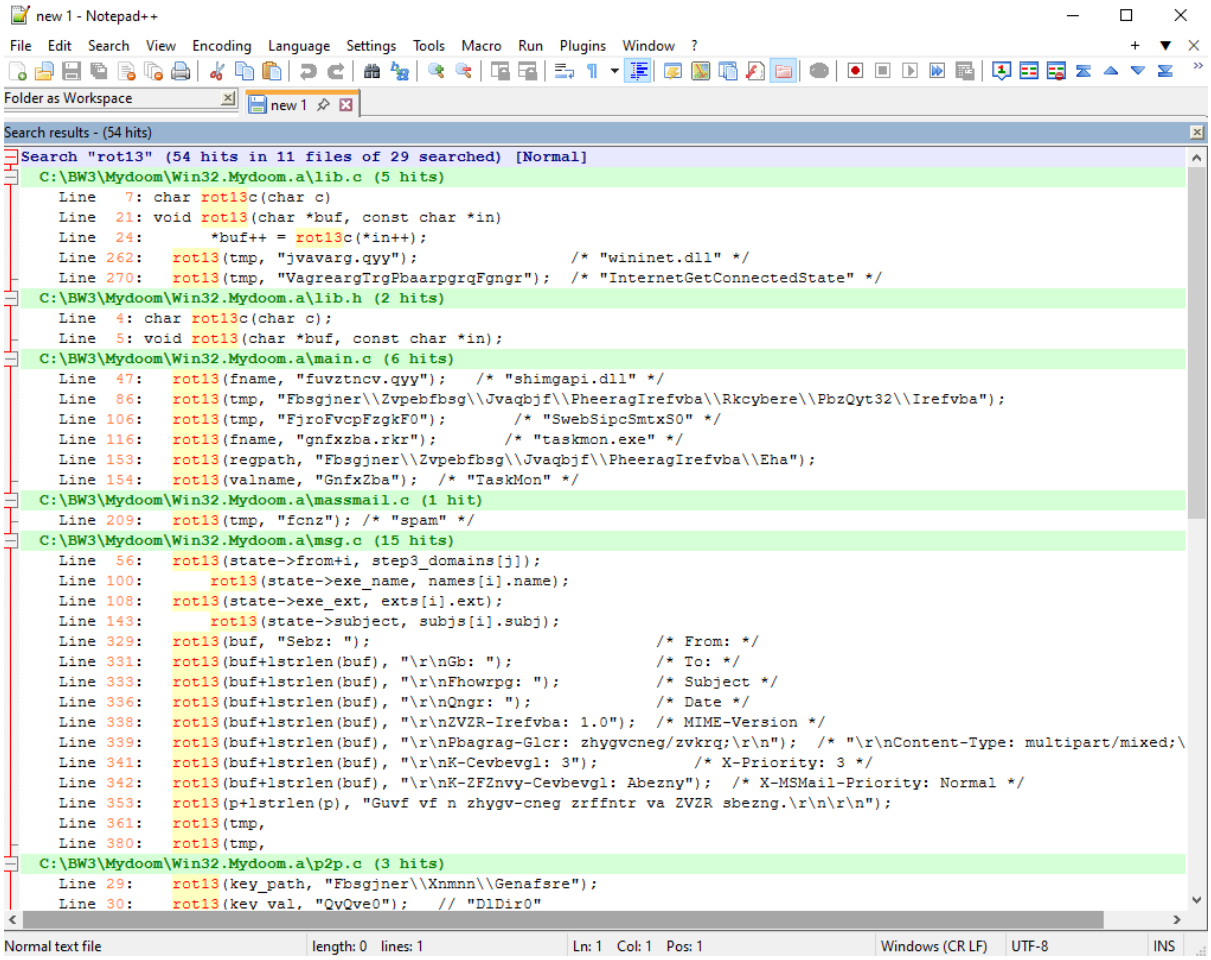
Per quanto riguarda invece le tecniche di **Evasione tramite Reimplementazione LibC (xstr*)**,

Le funzioni isolate come xstrstr, xstrrchr, xstrchr sono rimpiazzi diretti delle funzioni standard della libreria C per la manipolazione delle stringhe.

La valutazione Architetturale mostra che l'autore ha scelto di scrivere le proprie funzioni di stringa invece di importarle.

**Questa è una chiara tecnica piuttosto sofisticata di ottimizzazione ed evasione. Riduce le dipendenze esterne** (es. la libreria msvcrt.dll), **minimizza la dimensione del binario finale** e, soprattutto, riduce drasticamente l'Import Address Table (IAT).

Meno chiamate API documentate ci sono nella IAT, meno firme statiche gli analisti AV possono utilizzare per il rilevamento.




---

# Evasione dei Sistemi di Rilevamento di Rete (IDS/IPS)

Il traffico di rete viene pre-offuscato nel binario.

- **Comandi SMTP Nascosti (`xsmtp.c`):** Le direttive fondamentali del protocollo SMTP, come `EHLO`, `MAIL FROM` e `RCPT TO` (righe 206-217), non esistono in chiaro nel codice compilato. Vengono assemblate in memoria solo millisecondi prima dell'invio sul socket per ingannare le firme dei firewall perimetrali.

```
Line 206:   rot13(fmt, "RUYB %f\r\n");  /* EHLO %s */
Line 209:       rot13(fmt, "URYB %f\r\n");  /* "HELO %s\r\n" */
Line 214:   rot13(fmt, "ZNVY SEBZ:<%f>\r\n");   /* "MAIL FROM:<%s>\r\n" */
Line 217:   rot13(fmt, "EPCG GB:<%f>\r\n");     /* "RCPT TO:<%s>\r\n" */
```

- **Header MIME Forgiati (`msg.c`):** Tutti gli identificatori delle e-mail di spam (`From:`, `To:`, `Subject:`, `MIME-Version:`) sono cifrati per rendere la profilazione statica del payload estremamente complessa.

```
Line 143:       rot13(state->subject, subjs[i].subj);
Line 329:   rot13(buf, "Sebz: ");                           /* From: */
Line 331:   rot13(buf+lstrlen(buf), "\r\nGb: ");            /* To: */
Line 333:   rot13(buf+lstrlen(buf), "\r\nFhowrpg: ");       /* Subject */
Line 336:   rot13(buf+lstrlen(buf), "\r\nQngr: ");          /* Date */
Line 338:   rot13(buf+lstrlen(buf), "\r\nZVZR-Irefvba: 1.0"); /* MIME-Version */
Line 339:   rot13(buf+lstrlen(buf), "\r\nPbagrag-Glcr: zhygvcneg/zvkrq;\r\n");  /* "\r\nContent-Type: multipart/mixed;\r\n" */
Line 341:   rot13(buf+lstrlen(buf), "\r\nK-Cevbevgl: 3");          /* X-Priority: 3 */
Line 342:   rot13(buf+lstrlen(buf), "\r\nK-ZFZnvy-Cevbevgl: Abezny");  /* X-MSMail-Priority: Normal */
Line 353:   rot13(p+lstrlen(p), "Guvf vf n zhygv-cneg zrffntr va ZVZR sbezng.\r\n\r\n");
Line 361:   rot13(tmp,
Line 380:   rot13(tmp,
```

# Analisi della Ricerca “SCO”

```
Line   7: #define SCO_SITE_ROT13 "jjj.fpb.pbz"  /* www.sco.com */
Line  68:   rot13(buf,
Line  75:       "Ubfg: " SCO_SITE_ROT13 "\r\n"
Line 101:   rot13(buf, SCO_SITE_ROT13);
```

La ricerca della stringa “SCO” ha evidenziato la definizione:

**#define SCO_SITE_ROT13 "jjj.fpb.pbz"  /* www.sco.com */**

Il dominio è memorizzato in forma ROT13, e successivamente decodificato tramite la funzione `rot13()` presente nel codice. Questo indica che **il malware utilizza una tecnica di offuscamento semplice per nascondere riferimenti a domini esterni all’interno del sorgente**.

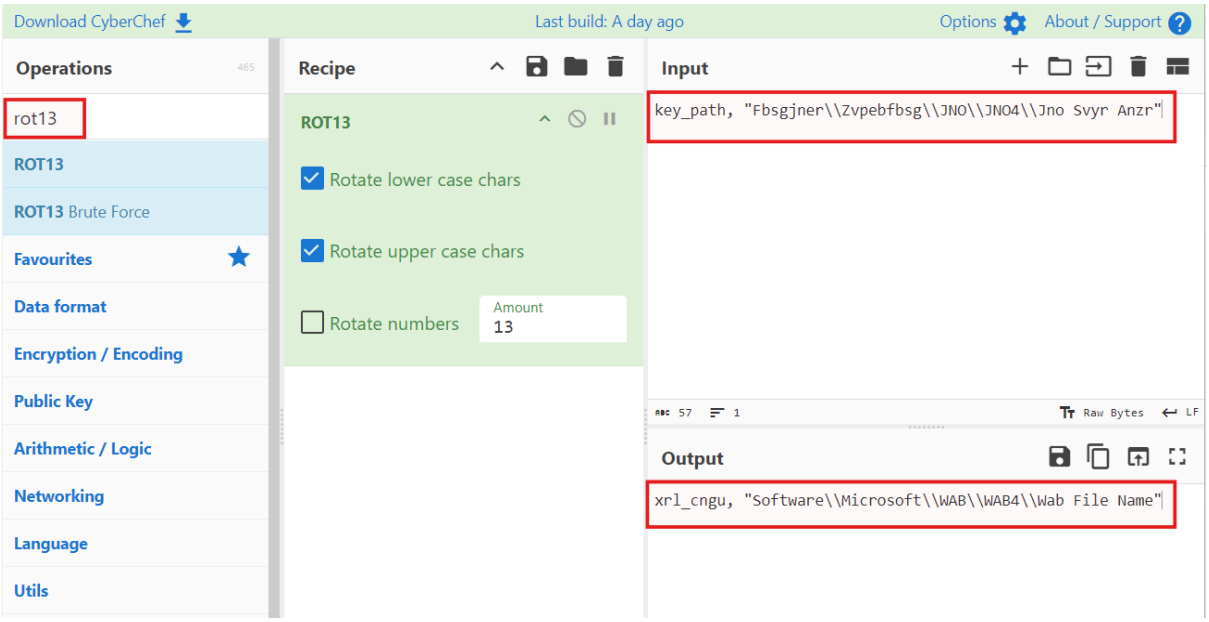
La presenza del dominio `www.sco.com` (decodificato) **è coerente con il comportamento storico di Mydoom, che includeva componenti mirate contro il sito SCO, utilizzato come target per attività di tipo DDoS**.

La presenza della costante `SCO_SITE_ROT13` **dimostra l’uso di offuscamento tramite ROT13 per nascondere domini target all’interno del codice**. Questo conferma l’implementazione di meccanismi di comunicazione o attacco verso host specifici, evidenziando una componente di attacco diretto integrata nel malware.

---

# Analisi delle Stringhe Offuscate con CyberChef (ROT13)

Durante l’analisi statica del codice **è stata individuata una stringa apparentemente offuscata tramite algoritmo ROT13**. Per verificarne il contenuto reale **è stato utilizzato CyberChef, applicando l’operazione di decodifica ROT13.**




Procedura eseguita:

1. Inserimento della stringa offuscata nel campo Input di CyberChef:

   **key_path, "Fbsgjner\\Zvpebfbsg\\JNO\\JNO4\\Jno Svyr Anzr"**

2. Applicazione dell'operazione ROT13 con:
   - Rotate lower case chars ✔
   - Rotate upper case chars ✔
   - Amount: 13
3. Analisi del risultato nel campo Output.

Risultato della decodifica:

**xrl_cngu, "Software\\Microsoft\\WAB\\WAB4\\Wab File Name"**

---

## Interpretazione Tecnica

La stringa decodificata fa riferimento al percorso:

**Software\Microsoft\WAB\WAB4\Wab File Name**

Il percorso WAB (Windows Address Book) è associato alla rubrica di Outlook/Windows Mail.

Questo conferma che **il malware accede alla rubrica di sistema per estrarre indirizzi email, che vengono successivamente utilizzati dal modulo di mass-mailing per la propagazione del worm.**

L'utilizzo della codifica ROT13 nel sorgente dimostra una tecnica di offuscamento basilare impiegata per nascondere riferimenti sensibili (come chiavi di registro) e rendere meno immediata l'analisi del codice.

**L'analisi con CyberChef ha permesso di decodificare correttamente una stringa ROT13, rivelando un riferimento alla rubrica di Windows (WAB)**. Questo rafforza l'evidenza che Mydoom implementa meccanismi di raccolta automatica degli indirizzi email dal sistema locale per alimentare la propria propagazione.

---

# Valutazione Variante e Possibili Aggiornamenti

**L'analisi del codice sorgente, integrata con la decodifica delle stringhe offuscate tramite CyberChef (ROT13)**, ha consentito di comprendere in modo più completo il comportamento reale del malware. In particolare, **la decodifica ha rivelato riferimenti alla chiave di registro** `Software\Microsoft\WAB\WAB4\Wab File Name`, **confermando l'accesso alla rubrica di Windows per l'estrazione automatica degli indirizzi email**.

Alla luce di quanto emerso, è possibile individuare diversi elementi potenzialmente modificabili in una variante evoluta del malware:

- I domini hardcoded (come il target SCO) potrebbero essere sostituiti con infrastrutture dinamiche, riducendo la tracciabilità statica.
- L'offuscamento ROT13, facilmente reversibile tramite strumenti come CyberChef, potrebbe essere rimpiazzato con tecniche più robuste per ostacolare l'analisi statica.
- Il motore SMTP interno, sebbene completo, potrebbe essere aggiornato per adattarsi ai moderni sistemi di filtraggio email.
- La gestione degli allegati MIME e ZIP potrebbe essere resa più variabile e meno prevedibile.
- I meccanismi di fallback DNS e generazione host (`smtp.<dominio>`, `mx1.<dominio>`) potrebbero essere resi meno deterministici.

Eventuali aggiornamenti in questi ambiti indicherebbero un'evoluzione dell'architettura operativa del malware verso maggiore evasione e adattabilità.

# In sintesi

La valutazione delle possibili modifiche si basa sull'**identificazione dei componenti più datati o facilmente analizzabili — come SMTP diretto, domini hardcoded, ROT13 e allegati prevedibili** — e sulla comprensione del loro funzionamento reale **grazie anche all'uso di CyberChef per la decodifica delle stringhe offuscate.**

L'analisi ha quindi permesso di distinguere tra ciò che il codice mostra superficialmente e ciò che effettivamente esegue una volta decodificato, fornendo una visione più accurata del comportamento del malware.

---

# Conclusione

L'analisi statica del codice sorgente di Mydoom, condotta in ambiente FlareVM e supportata dalla decodifica ROT13 tramite CyberChef, ha evidenziato una **struttura modulare orientata alla propagazione massiva tramite SMTP**. Il malware implementa direttamente i comandi di invio email, accede alla rubrica di Windows (WAB) per l'estrazione degli indirizzi e genera dinamicamente allegati malevoli in formato MIME/ZIP.

L'utilizzo di tecniche di offuscamento basilari e **la presenza di target hardcoded confermano un design tipico dei worm di prima generazione**. Tuttavia, l'**analisi ha anche permesso di individuare potenziali aree di evoluzione, evidenziando come una variante moderna potrebbe adottare meccanismi di evasione più sofisticati** e infrastrutture meno statiche.

Nel complesso, l'attività ha consentito di comprendere in modo approfondito l'architettura interna del worm e di rafforzare le competenze di analisi statica e interpretazione delle tecniche di offuscamento, fornendo elementi utili per la detection e la mitigazione di minacce analoghe.