BW III - Analisi MyDoom

Informazioni generali

Nome malware: MyDoom
Tipo: Worm
Data di comparsa: 26/01/2004
Sistema operativo target: Windows
Modalità diffusione: E-mail, reti peer-to-peer (KaZaA)

Introduzione

Mydoom è uno dei worm informatici più devastanti della storia, comparso per la prima volta nel gennaio 2004. La sua diffusione ha avuto un impatto significativo a livello globale, causando rallentamenti nelle reti aziendali e riducendo la velocità di Internet fino al 50% in alcuni casi. Il worm ha colpito i sistemi operativi Windows, propagandosi principalmente attraverso e-mail di phishing e reti peer-to-peer.
Una volta attivato, Mydoom si installava automaticamente nel sistema e iniziava a scansionare la rubrica di Microsoft Outlook alla ricerca di nuovi destinatari a cui inviare copie di sé stesso. Il worm sfruttava tecniche di spoofing per falsificare gli indirizzi mittenti, rendendo complesso il tracciamento dell'origine dell'infezione. Inoltre, apriva una backdoor sulla porta 3127/TCP, permettendo accessi remoti al sistema infetto e rendendolo vulnerabile a ulteriori attacchi informatici. Questa funzionalità ha favorito il controllo remoto dei dispositivi compromessi, aumentando il livello di minaccia per utenti e aziende.
Oltre alla sua capacità di propagazione, Mydoom è stato progettato per lanciare attacchi DDoS contro obiettivi specifici. Questi attacchi dimostrano la natura distruttiva del worm, che non si limitava solo a infettare sistemi ma mirava a danneggiare specifici bersagli attraverso attacchi coordinati.
Dal punto di vista della sicurezza informatica, Mydoom utilizzava diverse tecniche di evasione per eludere i sistemi di protezione. Si nascondeva nei processi di Windows per evitare il rilevamento da parte degli antivirus e cancellava i propri file temporanei per limitare le tracce della sua presenza.
L'impatto economico di Mydoom è stato stimato in oltre 38 miliardi di dollari, a causa dei danni provocati alle infrastrutture informatiche e della perdita di produttività dovuta ai rallentamenti di rete. Ancora oggi, alcune varianti di Mydoom continuano a circolare.

Analisi

Per analizzare il malware MyDoom siamo partiti dal codice principale **main.c**.
Tramite questa analisi abbiamo scoperto le sue funzioni principali:

1. **Avvio**
2. **Persistenza e autoriproduzione.**
3. **Diffusione tramite P2P (Kazaa)**
4. **Diffusione tramite email.**
5. **Esecuzione attacchi DDoS verso un sito (SCO.com)**
6. **Creazione backdoor**
7. **Offuscamento**
8. **Controllo temporale**
9. **Debug e visualizzazione**

Passiamo ora ad analizzare nel dettaglio queste funzioni e il modo in cui agisce il malware.

1. Avvio del Malware

L'avvio del malware e di tutte le sue componenti avviene grazie alla funzione **WinMain()**.
Si tratta del punto d'ingresso principale per un'applicazione Windows. Nel contesto del malware MyDoom, questa funzione svolge un ruolo cruciale nell'inizializzazione e nell'esecuzione delle attività dannose, orchestrando le operazioni fondamentali per l'infezione, la persistenza e la diffusione. In particolare:

- **xrand_init()**: Inizializza un generatore di numeri pseudo-casuali, probabilmente utilizzato per offuscare il comportamento del malware.
- **wsa_init()**: Inizializza la libreria Winsock (Windows Sockets API), fondamentale per la comunicazione di rete.
- **sync_main()**: Funzione principale che orchestra le varie operazioni del malware, richiamando tutte le sue funzionalità

2. Installazione, persistenza e autoriproduzione

Questi obiettivi del malware vengono raggiunti grazie alle seguenti funzioni principali, richiamate da **sync_main()**:

- **sync_check_frun()**: Verifica se il malware è in esecuzione per la prima volta controllando specifiche chiavi di registro offuscate tramite ROT13. Se è il primo avvio, imposta **first_run=1** e la crea in sia in **HKEY_LOCAL_MACHINE** che in **HKEY_CURRENT_USER**, garantendo la persistenza dell'informazione sull'avvio.
- **sync_install()**: Copia il malware in una directory di sistema (System32) o temporanea (Temp) con un nome offuscato (taskmon.exe) per confondere l'utente e gli antivirus.
- **sync_startup()**: Modifica il registro **Run** di Windows per garantire l'esecuzione automatica ad ogni avvio del sistema.

3. Diffusione tramite P2P

MyDoom è in grado di diffondersi utilizzando il client di condivisione file Kazaa. Il programma cerca di copiare se stesso in una cartella associata a Kazaa per diffondersi ulteriormente tramite la rete P2P (peer-to-peer).
Per fare ciò, il codice principale **main.c** utilizza la funzione **p2p_spread()**, la quale rappresenta il punto di avvio del processo di diffusione.
Essa ottiene il percorso dell'eseguibile tramite **GetModuleFileName** e lo passa alla funzione **kazaa_spread** che svolge molteplici funzioni:

- Definisce un array di nomi di file (**kazaa_names**) che il malware utilizzerà per copiare se stesso.
- Recupera il percorso di installazione di Kazaa dal registro di Windows. Il percorso specifico è nella chiave **Software\Kazaa\Transfer** sotto **HKEY_CURRENT_USER**.
- Se il percorso è valido, la funzione aggiunge uno dei nomi predefiniti da **kazaa_names** al percorso e crea un file che ha una delle estensioni seguenti: .exe, .scr, .pif, o .bat. Si tratta di tipi di file che possono sembrare legittimi, aumentando la probabilità che l'utente esegua il file infetto accidentalmente.

4. Diffusione tramite email

Un altro metodo di diffusione è la capacità di inviare grandi quantità di e-mail spam al fine di infettare altre macchine, utilizzando le funzioni **massmail_init** e **massmail_main_th**.

Queste funzioni, richiamano a loro volta diverse funzioni che gestiscono e ottimizzano l'invio di queste e-mail.

- **Gestione della coda di email**:
  - Il codice definisce una struttura di coda `mailq_t`, usata per tenere traccia delle email da inviare. La coda è gestita tramite la variabile globale **massmail_queue**.
  - Tramite la funzione **massmail_addq**, aggiunge elementi alla coda rispettando diversi parametri. Ad esempio, se l'indirizzo è già presente nella coda viene scartato.

- **Filtraggio e validazione indirizzi email**
  - **cut_email**: Estrae e normalizza un indirizzo e-mail valido da un input.
  - **email_check2**: Verifica se un indirizzo è valido, considerando lunghezza, dominio, ecc.
  - **email_filtdom** e **email_filtuser**: Confrontano il dominio e l'username con liste di esclusione per bloccare l'invio di email a domini noti di aziende antivirus e di istituzioni governative.
  - **email_filter**: Applica diversi filtri e scarta gli indirizzi che non li superano.

- **Generazione indirizzi e-mail**
  - **mm_gen**: Crea e-mail falsificate, selezionando un nome casuale da una lista (**gen_names**) e combinandolo con un dominio estratto da uno degli indirizzi nella coda di invio.

- **Gestione dei DNS per l'invio delle email**:
  Poiché il malware non usa un server SMTP legittimo, deve trovare autonomamente i server di posta dei destinatari, utilizzando funzioni come **mmdns_getcached**, **mmdns_addcache** e **mm_get_mx**.

- **Invio delle email**
  - **mmsender**: Estratto il dominio del destinatario, trova i relativi mail server. Genera il contenuto del messaggio tramite **msg_generate** ed esegue l'invio chiamando **smtp_send**.
  - **mmsender_th**: Permette l'esecuzione in thread separati.

- **Schedulazione delle email**
  - **massmail_main**: Questa funzione gestisce la coda di invio delle e-mail. Monitora continuamente lo stato della coda e avvia i thread per l'invio delle e-mail (**mmsender_th**). Infatti il sistema è progettato per essere altamente asincrono, con thread separati che gestiscono l'invio delle e-mail per evitare colli di bottiglia monitorati tramite la funzione **mmshed_run_threads**.
    Le e-mail inviate sono gestite attraverso uno scheduler, che rimuove gli indirizzi scaduti dalla coda (**mmshed_rmold**).
  - **massmail_init**: Inizializza la coda di email e imposta il contatore dei thread in esecuzione.
  - **massmail_main_th**: avvia un thread separato e chiama **massmail_main()**, assicurando che il malware continui a inviare email anche se il thread principale viene terminato.

5. Esecuzione attacchi DDoS verso un sito (SCO.com)

La capacità di eseguire un attacco DDoS verso un sito specifico, si deve alla funzione **payload_sco()**.
Tale funzione, verifica la data di attivazione tramite **sco_date** e, se è il momento giusto, chiama la funzione **scodos_main()**. Quest'ultima fa riferimento al codice **sco.c** , quindi analizzandolo possiamo scoprire le sue principali caratteristiche:

- La funzione **connect_tv** crea una connessione a un server (in questo caso, www.sco.com) utilizzando un socket TCP. Se la connessione non riesce immediatamente, viene effettuato un timeout o un ritardo per riprovare.
- La funzione **scodos_th** crea più thread (64 thread) che ogni volta cercano di connettersi al server di destinazione (in questo caso www.sco.com) e inviano una richiesta HTTP di tipo GET.
- La funzione **scodos_main** gestisce il ciclo di vita dell'attacco. Verifica se la macchina è online, risolve l'indirizzo IP di www.sco.com tramite **gethostbyname**, quindi avvia più thread per inviare richieste HTTP al sito.

Il codice cerca di fare il "flood" del server con richieste in modo tale da sovraccaricarlo e potenzialmente causare un'interruzione del servizio.

6. Creazione backdoor

Il worm Mydoom stabilisce una comunicazione con il proprio server di comando e controllo (C&C) attraverso una backdoor che crea sul sistema infetto.
Per fare ciò utilizza la funzione **payload_proxy** che si occupa di estrarre, scrivere su disco e caricare una DLL malevola chiamata **shimgapi.dll**, utilizzata per eseguire codice dannoso.
Vediamo nel dettaglio come funziona:

- Prova a salvare il file nella cartella di sistema di Windows (C:\Windows\System32). Se non riesce, prova a salvarlo nella cartella temporanea dell'utente (C:\Users\NomeUtente\AppData\Local\Temp).
- Una volta individuato il percorso e controllato che finisca con **** per evitare errori, aggiunge il nome del file **shimgapi.dll** al percorso, ottenendo qualcosa come "**C:\Windows\System32\shimgapi.dll**"
- Cerca di creare un file **shimgapi.dll** nel percorso scelto. Se il file esiste già, imposta lo stato a 2 (già installato) e salva il percorso.
- I dati della DLL presenti nel malware sono criptati, pertanto vengono decriptati con **decrypt1_to_file()**. Successivamente i dati vengono scritti nel file **shimgapi.dll** e, una volta fatto, lo stato viene impostato ad 1 per indicare che la scrittura è avvenuta con successo.
- Se la DLL è stata creata correttamente, viene caricata in memoria (**LoadLibrary**) e lo stato viene impostato a 2 per indicare che è attiva.

Il file **shimgapi.dll** funge da backdoor, consentendo agli aggressori di accedere e controllare il computer infetto da remoto. La backdoor si associa alla porta TCP 3127, permettendo potenzialmente di ricevere aggiornamenti o eseguire programmi arbitrari sul sistema compromesso.

Inoltre, il worm modifica una chiave del registro di sistema per caricare il file **shimgapi.dll** come thread del processo Explorer, garantendo così che la backdoor sia attiva ogni volta che il sistema operativo viene avviato.

7. Offuscamento

La principale tecnica di offuscamento utilizzata da MyDoom consiste in una funzione di cifratura **ROT13**. Si tratta di un tipo di cifratura simmetrica in cui ogni lettera dell'alfabeto viene sostituita dalla lettera che si trova 13 posizioni più avanti rispetto alla sua posizione originale. Se una lettera è già nella metà dell'alfabeto, viene fatta "ripartire" dall'inizio.
Inoltre:

- Se il carattere è una lettera maiuscola (A-Z), cerca la sua posizione nell'array u[] (che contiene tutte le lettere maiuscole).
- Se il carattere è una lettera minuscola (a-z), cerca la sua posizione nell'array l[] (che contiene tutte le lettere minuscole).
- Se il carattere non è una lettera, viene restituito invariato (ad esempio numeri o simboli).

Nel malware MyDoom, **ROT13** è usato per offuscare stringhe chiave, rendendo meno evidente il loro significato. Indirizzi IP, URL e comandi vengono codificati per evitare di essere individuati facilmente dagli analisti o dai software antivirus. Durante l'esecuzione, il malware decodifica dinamicamente le stringhe, rivelando le informazioni reali senza lasciarle in chiaro nel file.
Ad esempio, se MyDoom comunica con un server remoto su “example.com”, nel codice la stringa potrebbe apparire come “rkcbcr.zpb”. Il malware, una volta attivo, applica **ROT13** inversamente per ottenere l’indirizzo originale e stabilire la connessione. Questo metodo ostacola il rilevamento statico, poiché le stringhe cifrate non risultano immediatamente sospette.

8. Controllo temporale

Il malware effettua un controllo temporale in diversi modi. Uno di questi, lo abbiamo esaminato precedentemente durante l’analisi dell’attacco DDoS (punto **5**).
In quel caso, la funzione **payload_sco** effettuava un controllo della data di attivazione tramite **sco_date** prima di richiamare la funzione **scodos_main()**.
Un altro modo è tramite la funzione **sync_checktime()** che verifica se l'orario attuale ha superato una data memorizzata in una struttura **sync_t**.
Per farlo, usa **GetSystemTimeAsFileTime** per ottenere l'ora corrente in formato **FILETIME** e converte, nello stesso formato, la data di riferimento “**sync->termdate**” (data di termine) con **SystemTimeToFileTime**. Il confronto avviene prima sui valori **dwHighDateTime**: se l'orario attuale è maggiore, la funzione restituisce 1; se è minore, restituisce 0. Se uguali, confronta **dwLowDateTime** con la stessa logica.

9. Debug e visualizzazione

MyDoom utilizza la funzione **sync_visual_th** per integrare elementi di “distrazione” per l’utente. Tale funzione genera e visualizza un file temporaneo con Notepad.

- Inizia recuperando il percorso della directory temporanea con **GetTempPath**.
- Crea un nuovo file nella cartella temporanea con nome "Message". Se la creazione fallisce, esce dalla funzione.

- Un loop di 4096 iterazioni riempie un buffer con caratteri pseudo-casuali (valori ASCII tra 16 e 254) e salti di stringa. Il buffer viene poi scritto nel file.
- Costruisce il comando per aprire il file con Notepad e crea un processo per eseguire Notepad in primo piano.
- Attende che l'utente chiuda Notepad e successivamente elimina il file temporaneo e termina il thread.