ANALISI FORENSE MYDOOM

Analisi funzioni di propagazione

MyDoom si propaga attraverso diversi metodi:

1. **Email** - E’ il metodo principale di diffusione.

   Il worm:

- Invia copie di sè stesso come allegato email
- Utilizza oggetti email ingannevoli
- Finge di essere un messaggio di errore del server mail

2. **Network sharing P2P** - Si diffonde attraverso le reti peer-to-peer Kazaa, condividendo copie con nomi accattivanti
3. **Backdoor** - Installa una backdoor sulla porta 3127 che permetteva agli attaccanti di controllare remotamente i sistemi infetti
4. **Auto-replicazione** - Cerca indirizzi email nei file del computer infetto per propagarsi ulteriormente
5. **Ingegneria sociale** - Utilizza tecniche di social engineering nei messaggi email per ingannare gli utenti e farli aprire gli allegati infetti

1. Dall’analisi del codice sorgente abbiamo rilevato diverse librerie e funzioni che sono utilizzate per la diffusione del malware mediante email in particolare risulta molto rilevante il codice del sorgente **massmail.c**

ARCHITETTURA DEL SISTEMA DI MASS MAILING

[Scanner Email] --> [Coda Email] --> [Sistema Invio] --> [Server SMTP]

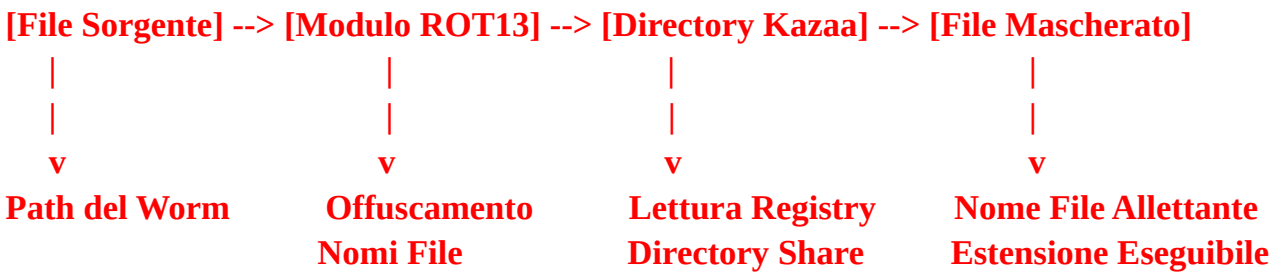
|	**|**	**|**	**|**
|	**|**	**|**	**|**
v	**v**	**v**	**v**
Raccolta Indirizzi	**GestioneQueue Priorità Filtri**	**Gestione Invii Multi-thread Cache DNS**	**Delivery Email**

Il sistema analizzato implementa un'architettura modulare per l'invio massivo di email, con i seguenti componenti principali:

1. Sistema di Gestione Code
    - Implementa una coda prioritaria degli indirizzi destinatari
    - Traccia lo stato di ogni email (non processata/in processo/completata)
    - Implementa timeout automatici dopo 2 ore
    - Limite massimo di 4096 email in coda
2. Sistema di Filtro
    - Validazione sintattica degli indirizzi email
    - Filtri per lunghezza username (2-24 caratteri) e dominio (6-42 caratteri)
    - Blacklist di domini sensibili (gov, mil, security vendors)
    - Filtri per indirizzi amministrativi comuni
3. Cache DNS
    - Cache di 256 entry per record MX
    - Sistema di reference counting
    - Gestione automatica scadenza cache
    - Ottimizzazione query DNS
4. Sistema di Invio
    - Architettura multi-thread (4 thread paralleli)
    - Gestione automatica connettività
    - Sistema di priorità (.edu priorità aumentata)
    - Generazione automatica mittenti plausibili

2. Il sorgente p2p.c è molto rilevante per comprendere come il malware si diffondeva attraverso le reti peer-to-peer

SISTEMA DI PROPAGAZIONE P2P



Il codice implementa un sistema di propagazione attraverso reti peer-to-peer (P2P), specificamente Kazaa, con le seguenti caratteristiche:

1. Sistema di Mascheramento File
   - Array di nomi predefiniti cifrati con ROT13
   - Nomi progettati per apparire come software desiderabile (crack, patch)
   - Randomizzazione delle estensioni eseguibili (.exe, .scr, .pif, .bat)
   - Offuscamento dei nomi reali per eludere il rilevamento
2. Integrazione con Kazaa
   - Accesso al registro di Windows per localizzare l'installazione
   - Identificazione della directory di condivisione
   - Utilizzo delle API di Windows per operazioni su file
   - Copia automatica nelle cartelle condivise
3. Tecniche di Offuscamento
   - Utilizzo di ROT13 per cifrare stringhe sensibili
   - Nomi file apparentemente legittimi
   - Variazione casuale delle estensioni
   - Mantenimento dei metadati originali del file

Il sistema mostra una progettazione finalizzata alla propagazione automatica attraverso reti P2P, sfruttando la tendenza degli utenti a scaricare software apparentemente legittimo. L'implementazione include tecniche di offuscamento per evitare il rilevamento e massimizzare la diffusione.

Questa struttura evidenzia l'intento di propagazione malevola attraverso l'ingegneria sociale e lo sfruttamento delle reti di condivisione file.

3. I sorgenti xproxy.c e client.c sono rilevanti per comprendere la creazione della backdoor e la persistenza nei sistemi vittima.

SISTEMA BACKDOOR SOCKS4

[Client] --> [Proxy SOCKS4] --> [Sistema Vittima]

| | |

| | |

v v v

Invio File Gestione Conn. Esecuzione File

+ Comandi + Relay Dati + Persistenza

Il codice implementa una backdoor basata su protocollo SOCKS4 con le seguenti componenti:

1. Server Proxy (xproxy.c)
    - Servizio proxy SOCKS4 sulle porte 3127-3198
    - Supporto comandi speciali (byte magico 133)
    - Auto-installazione nel registro di Windows
    - Multi-threading per gestione connessioni
    - Persistenza attraverso chiavi di registro Explorer
2. Client di Controllo (client.c)
    - Interfaccia di comando per connessione al proxy
    - Upload ed esecuzione remota di file
    - Gestione comunicazione con il server
    - Supporto trasferimento dati binari
3. Caratteristiche Principali
    - Protocollo SOCKS4 come copertura
    - Offuscamento stringhe con ROT13
    - Avvio automatico con Windows
    - Supporto Windows 9x/NT
    - Gestione thread dinamica
4. Meccanismi di Evasione
    - Mascheramento come servizio legittimo
    - Utilizzo di protocollo standard
    - Rotazione porte in caso di blocco
    - Injection in processi di sistema

Il sistema mostra una progettazione finalizzata all'accesso remoto non autorizzato e alla persistenza sui sistemi compromessi.

4. Dopo aver ottenuto la persistenza il malware esegue delle scansioni sul sistema vittima per autoreplicarsi. Il sorgente scan.c è rilevante per capire in che modo si autoreplica.

SISTEMA DI HARVESTING EMAIL

[File Scanner] --> [Parser Email] --> [Filtro/Validazione] --> [Output Queue]

| | | |

| | | |

v v v v

Scansione File Directory/WAB | **Estrazione Email Pattern Matching** | **Normalizzazione Decodifica HTML** | **Invio a Queue Mass Mailer**

Il sistema implementa un harvester di indirizzi email con le seguenti componenti:

1. Sistema di Scansione
   - Scansione ricorsiva di directory (max 15 livelli)
   - Supporto per file di testo (.txt, .html, .php, .asp)
   - Parser specifico per Windows Address Book (WAB)
   - Scansione dei file temporanei Internet Explorer
2. Engine di Estrazione Email
   - Tabella di caratteri validi per email
   - Pattern matching per @ e domini
   - Supporto per vari formati (user@domain)
   - Gestione lunghezze minime (7 caratteri)
3. Decodifica e Normalizzazione
   - Conversione formati offuscati (@, (at), etc)
   - Decodifica HTML entities
   - Normalizzazione spazi e caratteri speciali
   - Rimozione markup HTML
4. Gestione Risorse
   - Limiti dimensione file (80KB-8MB)
   - Sleep tra scansioni (8 secondi)
   - Priorità thread ridotta
   - Freeze/unfreeze del sistema

Il sistema mostra una progettazione finalizzata alla raccolta massiva di indirizzi email da file locali, con particolare attenzione all'ottimizzazione delle risorse e alla gestione di formati diversi. Implementa tecniche sofisticate di parsing e decodifica per massimizzare il numero di indirizzi validi estratti.

Analisi tecniche di evasione della sicurezza

MyDoom usa diverse tecniche di evasione di sistemi di sicurezza:

1. **Tecniche di evasione:**
   - Modifica diverse chiavi di registro per l'avvio automatico
   - Blocca l'accesso a siti web di antivirus modificando HOSTS
   - Crea una backdoor SOCKS proxy sulla porta 3127
   - Disabilita Windows Update e Windows Security Center
   - Termina processi correlati alla sicurezza
2. **Tecniche Anti-analisi:**
   - Codice offuscato e criptato
   - Rilevamento di debugger e strumenti di analisi
   - Check per ambienti virtualizzati
   - Tecniche anti-dump della memoria
   - Auto-modifica del codice

Molte di queste tecniche vengono implementate mediante il codice già analizzato in xproxy e scan quindi ci concentreremo sull'analisi delle tecniche di offuscamento e sul modo in cui il codice viene criptato.

Analizziamo quindi il codice dei sorgenti rot13c.c e cripto1.c:

SISTEMI DI OFFUSCAMENTO

[ROT13] **[CRYPT1]**
| **|**
v **v**
Input -> Rotazione -> Output **Input -> XOR Stream -> Output**
13 pos. **Chiave variabile**

Il codice implementa due sistemi di offuscamento per nascondere informazioni sensibili:

1. ROT13 (rot13.c)
    - Cifrario a sostituzione semplice
    - Sposta ogni lettera di 13 posizioni nell'alfabeto
    - Caratteristiche:
        - Reversibile (applicarlo due volte ripristina il testo)
        - Mantiene case sensitiveness
        - Preserva caratteri non alfabetici
    - Usato principalmente per:
        - Offuscare stringhe nel codice
        - Nascondere path e chiavi di registro
        - Mascherare nomi di file e comandi
2. CRYPT1 (crypt1.c)
    - Cifrario XOR stream con chiave variabile
    - Caratteristiche:
        - Chiave iniziale: 0xC7
        - Evoluzione chiave: k = (k + 3 * (posizione % 133)) & 0xFF
        - Operazione su byte singoli
    - Usato per:
        - Cifrare file binari
        - Nascondere payload
        - Proteggere dati sensibili

Questi strumenti sono utilizzati per:

- Eludere rilevamento statico
- Nascondere stringhe significative
- Offuscare componenti malevole

Analisi comunicazione con i server di comando e controllo

Unendo le diverse tecniche analizzate in precedenza e analizzando gli altri sorgenti, è possibile stabilire come il malware gestisce la comunicazione con i server di comando e controllo:

Il MyDoom rappresenta uno dei primi esempi di malware con un'architettura di comando e controllo (C2) sofisticata per l'epoca. La sua infrastruttura di comunicazione era progettata per garantire robustezza e persistenza, caratteristiche che lo hanno reso uno dei worm più influenti nella storia del malware.

L'architettura C2 si basava principalmente su una backdoor SOCKS proxy implementata sulla porta 3127, con un sistema di backup sulla porta 1034. Questa ridondanza era fondamentale per mantenere il controllo della rete di macchine infette. Il sistema utilizzava comunicazioni TCP dirette, ma la vera innovazione stava nella sua natura decentralizzata, che evitava vulnerabilità legate a singoli punti di fallimento.

La comunicazione all'interno della rete MyDoom era particolarmente avanzata. Il malware implementava un'infrastruttura peer-to-peer per la distribuzione dei comandi, permettendo una gestione più resiliente della botnet. Le connessioni TCP dirette ai server C2 erano integrate con meccanismi di crittografia basilare e una funzionalità di port scanning che permetteva di identificare altri host infetti, creando così una rete auto-propagante.

Dal punto di vista funzionale, il sistema C2 era molto versatile. Poteva coordinare attacchi DDoS distribuiti, gestire il download di payload aggiuntivi e implementare aggiornamenti del malware stesso. Inoltre, raccoglieva e trasmetteva informazioni dettagliate sui sistemi infetti, permettendo agli attaccanti di mantenere una mappatura precisa della loro rete.

Per quanto riguarda l'evasione dei sistemi di sicurezza, MyDoom utilizzava diverse tecniche sofisticate. La rotazione degli indirizzi IP dei server C2 rendeva più difficile il blocco delle comunicazioni. Il traffico veniva offuscato per evitare il rilevamento, e l'utilizzo di porte comunemente aperte aiutava a confondere il traffico malevolo con quello legittimo. Una delle tecniche più efficaci era la modifica del file hosts per impedire agli antivirus di aggiornarsi.

La resilienza era garantita da diversi meccanismi di backup. Oltre alla rete P2P che fungeva da sistema di fallback, il malware utilizzava multiple porte di comunicazione e manteneva una lista hardcoded di server C2. Il sistema di auto-propagazione assicurava che la rete rimanesse attiva anche in caso di perdita di numerosi nodi.