Comparative Analysis of Botnet and Ransomware for Early Detection

Prasad Honnavalli B[1] (prasadhb@pes.edu), Ethadi Sushma[2] (sushmae@pes.edu), Aditya Rao[3] (adityarao1kiran@gmail.com), Varun Girimaji[4] (vgirimaji@gmail.com), Vrinda Girimaji[5] (girimajiv@gmail.com), and Achyuta Katta[6] (pes1ug20cs618@pesu.pes.edu)

Dept. of CSE,
PES University,
Bangalore

Abstract. Recent cyber threats highlight the formidable challenges posed by ransomware and botnets to critical infrastructure. Ransomware, holding data hostage for hefty payments, and botnets, orchestrating distributed attacks through compromised machines, demand distinct mitigation approaches. In this paper, we study ransomware and botnets, using WannaCry and Mirai as examples. We also look into the similarities and differences between the two classes of malware, as well as the techniques they employ. Additionally, this paper describes measures that can be used to prevent and mitigate these threats.

1 Introduction

Malware analysis is a critical field in cybersecurity that involves the identification, evaluation, and understanding of malicious software - more commonly referred to as malware. Malware is designed to cause harm to computer systems, steal sensitive information, and perhaps even disrupt network services. They can be introduced into systems through phishing, social engineering and software vulnerabilities.

The goal of malware analysis is to gain insight into how malware operates, its capabilities, and its potential impact on a system. Malware analysis can be performed using different techniques, including static and dynamic analysis. Static analysis involves examining the code or file without running it, while dynamic analysis involves executing the malware and monitoring its behavior in a controlled environment. To begin with, we performed static analysis on the *Mirai* botnet, followed by a static and dynamic analysis of the WannaCry ransomware. We then compared ransomware and botnets against each other, taking Mirai and WannaCry respectively as examples.

Malware analysis can be used in multiple scenarios, including, but not limited to,

- Incident Response: Identifying the cause and scope of a security breach in an organization's network.

– Forensics: Collecting and analysing digital evidence
– Threat Intelligence: Tracking the motives of cyber-criminals
– Assessment of Vulnerabilities: To discover vulnerabilities/bugs that can pose as a danger for a variety of applications.
– Detecting and removing malware that has infected an organization's computers.
– Testing antivirus software
– To document the malware and create reports.

Ransomware and botnets are major threats in modern cybersecurity, as evidenced by many high-profile attacks in recent years. This paper provides a detailed analysis of the behavior, resilience, and evolutionary characteristics of these malware types, using WannaCry and Mirai as case studies. We also explore the strategies and tactics used by ransomware and botnets to evade and adapt to existing security measures, and highlight both new and existing strategies that can be used to defend against them.

2 Botnets and A Case Study in Mirai

A botnet comprises individual machines referred to as "bots" or "zombies," with each bot executing software that carries out specific tasks upon receiving instructions from a command-and-control (often abbreviated as C2 or CNC) server. Botnets generate revenue by offering their services for rent to attackers looking to orchestrate distributed attacks. The nature of these distributed attacks ranges from click fraud to distributed denial-of-service (DDoS) attacks. [4].

2.1 Evolution of Botnets

Botnets emerged from the evolution of IRC bots, initially designed to prevent IRC servers from being shut down from lack of activity [1]. One of the earliest significant botnets emerged in 1999, utilizing the Sub7 remote access trojan [1]. Notably, starting from version 2.1, it became controllable via IRC.

Over time, botnets evolved away from the initial IRC command-and-control model and started utilizing HTTP, ICMP, and SSL protocols [1]. Instead of relying on hardcoded IP addresses, they began to utilize domains offered by bulletproof hosting companies to enhance their reliability [4].

2.2 Mirai: A Case Study

Mirai, initially detected in August 2016 by the malware research group MalwareMustDie [5], has been implicated in some of the most impactful DDoS attacks in recent history. Notable targets include the security blog KrebsOnSecurity and the DNS provider Dyn [5]. The source code of this botnet was publicly shared by one of its authors, known as "Anna-senpai," in a forum post [5] [6].

Working Mirai is primarily designed for DDoS attacks, and its focus on this particular attack type contributes to its compact footprint, making it well-suited for targeting IoT devices. Mirai opts for a unique approach to infecting devices by attempting to crack credentials through brute force from a predefined list of default usernames and passwords. This strategy remarkably proved highly effective, with 600,000 participating bots at the peak of Mirai's influence [7].

The following are the steps performed to infect a new device:

1. A bot 'A' discovers a new device 'B' on the network that has not been infected
2. A attempts to log in to B by trying each username-password pair from a list in succession

```
// scanner.c, line 124
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\
    x13", 10);                      // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54"
    , 9);                           // root     vizxv
// ...
```

3. If A is able to log in successfully with one of these username-password pairs, it sends B's details to the command and control server.

```
// scanner.c, line 605
else if (consumed > 0)
{
    char *tmp_str;
    int tmp_len;
    report_working(conn->dst_addr, conn->dst_port, conn->
        auth);
    // ...
}
```

4. The command-and-control server stores B's details and logs in to B. This is done by calling `NewBot` in `main.c`
5. The CNC server then determines the architecture of B, and loads the appropriate shellcode.
6. The shellcode fetches the malicious executable from the download server and runs it, turning the device into a bot.

The CNC server plays a pivotal role in overseeing both the network's bots and its users. Each user is assigned a maximum number of bots, which they cannot exceed. Users may also receive authorization to execute attacks simultaneously, while those without authorization must wait until their ongoing attack concludes and the cooldown period elapses before launching another one. Furthermore, the CNC server ensures that attacks directed at whitelisted targets are not carried out.

Subsequent versions The release of the source code of Mirai resulted in a wave of Mirai-based malware, all targeting different but related platforms. Notable among these are *Okiru*, which targets the ARC (Argonauts RISC Core) architecture and *Masuta* and *PureMasuta*, which uses the EDB 38722 D-Link router's exploit to enlist further devices [5].

3 Ransomware and A Case Study in WannaCry

The AIDS trojan, the first ransomware, was distributed by Joseph L. Popp at a World Health Organization conference via floppy disks, demanding a $189 ransom sent to a P.O. box in Panama [8]. In the 2010s, ransomware evolved, causing billions in damages with the rise of cryptocurrency. Companies, especially larger ones, often opt for discreet ransom payments to protect their reputation, prioritizing financial concerns over public disclosure.

A crucial moment in ransomware history was the widespread WannaCry attack on May 12, 2017, impacting numerous vulnerable Windows computers. Understanding the operational details of this malware is vital for preventing future attacks. Subsequent variants like Petya and NotPetya operate on the same foundational code, emphasizing the need to unravel the workings of these malicious entities.

In 2017, the WannaCry malware made a profound impact on a global scale by infiltrating hundreds of thousands of computers. Operating by encrypting files and demanding ransom payments in Bitcoin, its repercussions extended beyond government and telecommunication services, reaching critical sectors such as healthcare. The National Health Service (NHS) in the United Kingdom experienced severe consequences, with over 60 trusts significantly affected. This led to the diversion of ambulances, the rescheduling of non-essential surgeries, and widespread disruption within the healthcare system, intensifying the magnitude of the crisis [2].

Upon infiltration by the WannaCry ransomware, the user is confronted with a changed desktop background and the distinctive red window, signaling the encryption of their files. The ransom demands prompt the user to remit 300USD within three days or 600USD within six days to a specified Bitcoin wallet. Failing to pay the ransom in seven days results in user's files remaining encrypted forever.

3.1 WannaCry- How the original variant worked

Figure 3.1 illustrates WannaCry's infection initiation, commencing with a check for a suspicious URL (`https://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com`). If the connection succeeds, the program terminates; otherwise, the infection process proceeds. Subsequently, *mssecsvc.exe* is executed, triggering either encryption initiation (when number of arguments <2) or the creation of a payload for network scanning and infecting vulnerable computers

(when arguments >2). These processes are known as the Dropper and Infection/Worm mechanism respectively, and the network activity of the latter can be seen clearly in Wireshark as shown in Figure 2.

Upon the detection of less than 2 parameters, the dropper component is activated. During the execution of this function, a service named *mssecevc_2.0* is registered to avoid antivirus detection and removal. It masquerades as Microsoft Security Center 2.0 for evasion purposes. Subsequently, a program, *tasksche.exe*, is created, and a resource is extracted into it, executed with the */i* argument. This resource contains a zipped folder, and the password for extraction becomes apparent when examining the arguments associated with the unzip function.

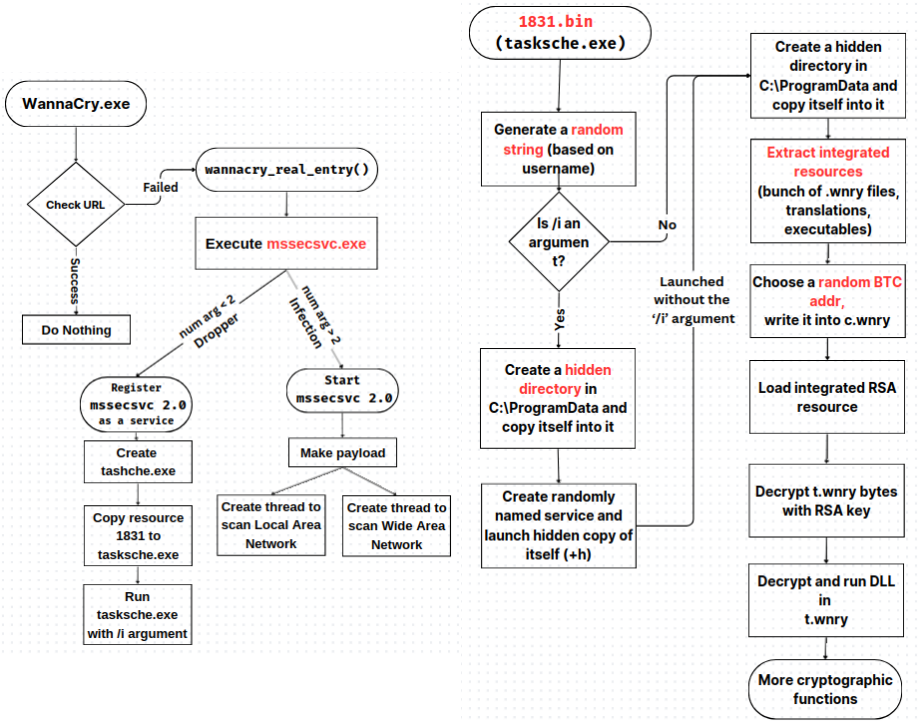

Fig. 1. Flow Chart of the WannaCry infection process

Upon scrutinizing the decompiled output of the extracted resource `tasksche.exe`, its functionality becomes apparent. Figure 3.1 provides the flowchart detailing this phase of execution. Primarily, it generates a random string based on the user's username and checks whether it was launched with the /i argument.

When initiated with the */i* argument, signifying WannaCry's first run on the computer, the program establishes a concealed directory, copies itself into it, and launches a service with the randomly generated name, along with the

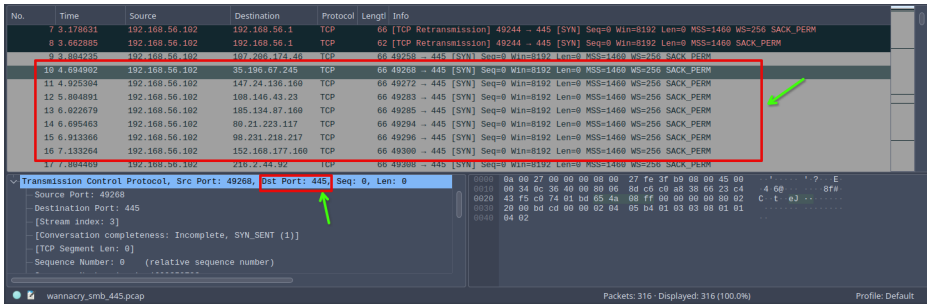

Fig. 2. Conducting scans to identify additional susceptible computers for infection, over port 445, followed the DNS lookup for the vulnerable URL. In the event of a failed connection attempt, the process of scanning for more computers to infect was initiated

Table 1. XIA Resources in the 1831 resource

File Name	Contents
`msg\m_*.wnry`	Ransom notes to be displayed in the alert window in different languages
`b.wnry`	Instructions for decrypting files
`c.wnry`	The Onion Router (TOR) addresses
`s.wnry`	TOR Browser's executable
`r.wnry`	README (ransom note FAQs)
`t.wnry`	DLL of the encrypted ransomware
`f.wnry`	demonstration that decryption works
`u.wnry`	decryptor file

hidden copy. This constitutes one of WannaCry's persistence techniques. Subsequently, the hidden copy is launched again, this time without the */i* argument. In this scenario, a hidden directory is created within `C:/ProgramData`, and a duplicate of itself is stored within this directory.

Embedded resources within this binary, including *.wnry* files, translations in multiple languages, and a couple of executables, are then extracted. Following this, a random Bitcoin address is selected from a list of three addresses and written into `c.wnry`. The subsequent phase involves cryptographic functions, wherein the RSA key decrypts `t.wnry`, containing numerous `.onion` addresses. The Table 1 shows the various XIA resources in the extracted binary.

In short, analyzing the initial WannaCry variant reveals key findings. The ransomware consists of multiple components, including a primary dropper housing an encrypter and a decryption application. The malware's worm functionality targets the SMB protocol using the EternalBlue exploit (on port 445) and checks for a Doublepulsar backdoor in the network. If Doublepulsar is absent, indicating a new infection, the EternalBlue exploit is executed for a fresh infection upon successful establishment of Doublepulsar on the target machine.

4 Similarities and Differences between Ransomware and Botnets

4.1 Shared Characteristics

Concealment of Presence Both WannaCry and Mirai employ evasion tactics, such as altering process names. Mirai enhances concealment by deleting the original executable, unlike WannaCry, which must persist after system reboots. Both malware generate random executable names to complicate detection.

Anti-debugging Measures Debuggers greatly simplify and streamline the process of dynamic analysis. Anti-debugging strategies are thus crucial to the longevity of malware, with all variants of both WannaCry and Mirai making use of anti-debugging measures in their code.

Both WannaCry and Mirai make use of code obfuscation to combat advanced static analysis tools, with Mirai employing this technique more extensively.

Evolutionary Patterns Mirai and WannaCry exhibit a shared trait in the frequent emergence of new variants post-initial outbreaks. Mirai's source code facilitates this, while WannaCry relies on a script for new versions.

Mirai's different variants often target different embedded RISC platforms, with some versions choosing to spread via a vulnerability over trying to guess credentials. Some variants split or encrypt strings in the executable to evade pattern-matching tools like YARA.

WannaCry, designed for x86-based Windows systems, discourages extensive porting efforts. Derivatives opt for subtle modifications, altering filenames and encrypted file extensions for evasion. Variations include *@WanaCryptor@.exe* and *@WannaDecrypt0r@.exe*.

4.2 Distinguishing Ransomware and Botnets

Goals It is very important for a botnet to spread and infect other systems, as the strength of a botnet is heavily dependent on the number of participating bots. A good botnet should also allow for multiple command-and-control servers for redundancy and load distribution.

Ransomware on the other hand does not have to try too hard to ensure its persistence as the users themselves will keep it on their system out of fear of losing their files. Command-and-control servers are also less critical and simpler to implement than in the case of ranosmware.

Infection and Spread Ransomware can be introduced into a network in various ways. The author could choose to send a malicious e-mail or direct the victim to a site that performs a drive-by download. There is also the option of having the software delivered through a botnet, an approach favoured by more recent malware like Clop and Ryuk.

The means for delivering botnets however is dependent on the target systems. Should the target system be an IoT device in the case of Mirai, other methods may need to be used. Mirai for example used the approach of sending Telnet requests to pseudo-random IP addresses to infect systems on other networks.

5 Prevention and Mitigation of Botnets- A Survey

Besides the obvious preventive solutions like updating system software and backing up important files, there are several measures that can be taken to prevent and mitigate the effects of malware. Some such measures are given below.

Antivirus Solutions Antivirus software scans files on a system, comparing their hashes with a threat database, and takes actions like quarantine or deletion upon detection. While not foolproof, modern antivirus integrates intrusion detection and evolving threat detection methods, serving as a vital initial defense.

Intrusion Detection and Prevention Systems These systems identify abnormal activity either on hosts or networks, reporting and, in the case of prevention systems, mitigating potential threats. They include rules tailored to detect ransomware and botnets, enhancing security.

YARA Pattern-Matching Tool YARA classifies malware based on binary patterns, offering rules for malware families. It aids in identifying and categorizing threats, providing a nuanced approach to threat analysis.

Infrastructure Design

Backups Maintaining isolated backups, especially critical records stored offline, acts as a potent defense against ransomware attacks targeting network drives.

Firewalls Modern firewalls are highly flexible and can be used to block very specific patterns of network traffic. Firewalls are a great defense against ransomware and botnets as they can restrict access to various ports on a potential victim that could be used by the malware to spread to it.

Better permissions system Restricting write access to backups stored on the network can greatly reduce the impact of a ransomware attack, making not paying the ransom and losing the files a sensible course of action.

5.1 Emerging Solutions

Semantic Analysis Semantic analysis focuses on analyzing the call graph of a malicious program, enhancing accuracy without increasing false positives. Approaches like the gSpan algorithm aid in identifying common subgraphs among malware samples, offering a sophisticated detection mechanism.

Software-Defined Networking (SDN) The use of SDN for ransomware detection and mitigation, as demonstrated in the case of WannaCry, involves writing applications for OpenFlow controllers. These applications analyze packets, detect malicious communication, and dynamically block infected hosts without an impact on the throughput of the connection.

6 Conclusion

In the aftermath of recent cyberattacks, it is evident that ransomware and botnets pose significant threats to critical infrastructure. Ransomware operates by seizing control of data and demanding substantial ransom payments, whereas botnets harness a multitude of compromised machines to orchestrate distributed attacks. Despite their shared disruptive potential, ransomware and botnets represent distinct threats, necessitating tailored mitigation strategies. Various established solutions exhibit considerable efficacy against these threats, with even more potent countermeasures emerging. Integrating these solutions into a robustly designed infrastructure serves as a formidable defense, safeguarding organizations and enabling uninterrupted operational continuity.

References

1. S. MacConnell, "The History Of Botnets: Everything You Need To Know", BusinessComputingWorld. https://businesscomputingworld.co.uk/the-history-of-the-botnet-part-1/ (Accessed Nov. 09, 2023)
2. Collier, R. NHS ransomware attack spreads worldwide. *CMAJ.* **189**, E786-E787 (2017), https://www.cmaj.ca/content/189/22/E786
3. "Sub7", Wikipedia. https://en.wikipedia.org/wiki/Sub7 (Accessed Nov. 09, 2023)
4. "Botnet", Wikipedia. https://en.wikipedia.org/wiki/Botnet (Accessed Nov. 09, 2023)
5. "Mirai (malware)", Wikipedia. https://en.wikipedia.org/wiki/Mirai_(malware) (Accessed Nov. 09, 2023)
6. Anna-senpai (alias). [Forum post]. https://github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.md (Accessed Nov. 09, 2023)
7. E. Bursztein, "Inside Mirai the infamous IoT Botnet: A Retrospective Analysis", [Personal blog]., https://elie.net/blog/security/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/ (Accessed Nov. 09, 2023)
8. Brewer, R. Ransomware attacks: detection, prevention and cure. *Network Security.* **2016**, 5-9 (2016), https://www.sciencedirect.com/science/article/pii/S1353485816300861
9. O'Kane, P., Sezer, S. & Carlin, D. Evolution of ransomware. *IET Networks.* **7**, 321-327 (2018), https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-net.2017.0207

10. Adamov, A. & Carlsson, A. The state of ransomware. Trends and mitigation techniques. (2017,9)
11. Ben, N., Biondi, F., Bontchev, V., Decourbe, O., Given-Wilson, T., Legay, A. & Quilbeuf, J. Detection of Mirai by Syntactic and Behavioral Analysis. (2018,10)
12. Sinanović, H. & Mrdovic, S. Analysis of Mirai malicious software. *2017 25th International Conference On Software, Telecommunications And Computer Networks (SoftCOM)*. pp. 1-5 (2017)
13. Algarni, S. Cybersecurity Attacks: Analysis of "WannaCry" Attack and Proposing Methods for Reducing or Preventing Such Attacks in Future. *ICT Systems And Sustainability*. pp. 763-770 (2021)
14. Ben Said, N., Biondi, F., Bontchev, V., Decourbe, O., Given-Wilson, T., Legay, A. & Quilbeuf, J. Detection of Mirai by Syntactic and Semantic Analysis. (2017), https://inria.hal.science/hal-01629040, working paper or preprint
15. Akbanov, M., Vassilakis, V. & Logothetis, M. Ransomware detection and mitigation using software-defined networking: The case of WannaCry. *Computers & Electrical Engineering*. **76** pp. 111-121 (2019), https://www.sciencedirect.com/science/article/pii/S0045790618323164
16. Symantec Symantec's initial blog. , https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/wannacry-ransomware-attack
17. Johnson, A. Endpoint Protection. , https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=b2b00f1b-e553-47df-920d-f79281a80269&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments
18. Malwaretech Blog about WannaCry. , https://www.malwaretech.com/2017/05/how-to-accidentally-stop-a-global-cyber-attacks.html
19. Wikipedia Wikipedia article- Mirai. , https://en.wikipedia.org/wiki/Mirai_(malware)
20. Wikipedia Wikipedia article- Botnet. , https://en.wikipedia.org/wiki/Botnet
21. Balaban, D. The 8 biggest botnets of all time. , https://cybernews.com/security/the-8-biggest-botnets-of-all-time/
22. Duncan, B. Emotet Malware summary. , https://unit42.paloaltonetworks.com/emotet-malware-summary-epoch-4-5/
23. Post, F. Uptime average. , https://forum.openwrt.org/t/survey-whats-your-uptime/56088/6