Email Worm Attacks

1st Aswal, V.
SFWE Graduate Student
vivekaswal@arizona.edu

2nd Destin, C.
ECE Graduate Student
cdestin@arizona.edu

3rd Hansen, B
ECE Graduate Student
brienhansen@arizona.edu

***Abstract*—Sending an Email is fast, easy and very flexible, therefore it has been a personal favorite for many users to communicate online. It is not only easy to use, it is sometimes a requirement for setting a billing account to pay one's mortgage, phone bill and even the basic utility bills. From being used in the workplace or for personal use, emails have also been the perfect method to launch an initial cyber attack on any victim without discrimination of age; teenagers or elderly, we are all potential victims to receiving a suspicious email containing a worm targeting our personal information or our hard earned income. In addition, the EmailListVerify states in its blog that "On average, a person, particularly in a professional setting, will receive a whopping 121 emails per day". With the knowledge of someone's email, an attacker can easily pose as someone else with the objective of stealing the victim's personal information through clever schemes. Email worms are particularly dangerous because they spread rapidly, often causing widespread network damage before they are detected. In some cases, companies can also be targets of a criminal cyber attack aimed directly at a company to retrieve the company's proprietary information or steal thousands of dollars. This type of attack is known also to the Federal Bureau of Investigation (FBI) who has listed this practice as a common fraud and scam. Despite the current effort in securing emails, attackers have developed clever methods for accessing a host's email information, an attacker can use bots, or host an api to bypass the security without raising any major concern and retrieve the information desired. This project will analyze the mechanisms of email worm attacks, and recreate an email worm attack scenario to demonstrate the threat that an email worm attack can cause to an individual and his computer; and the threat can spread to more computers in contact with the victim on the internet.**

***Index Terms*—Email Worms, Cyberattacks, Malware, Email**

I. Introduction

When it comes to criminals, the names Jesse James, Al Capone or Machine Gun Kelly are very famous to the common folks, however one may argue that Robert Tappan Morris and his launching the first computer worm on the entire country may be the one of the most important crimes committed in modern day cybersecurity and his crime is still having an impact 30 years later. In fact, the Federal Bureau of Investigation (FBI) wrote an article in 2018 which stated "the first Internet attack 30 years ago was a wake-up call for the country and the cyber age to come." This is in reference to the first computer worm known as the "Morris Worm", which was launched on the internet on November 2, 1988, and spread using an early form of emails. A worm is different from a virus, as it is able to replicate and spread itself without human intervention, at the time only 60,000 computers from less than 20 countries were using the internet, however, this attack has encouraged copycats in creating more sophisticated worm attacks. Modern day attackers take advantage of the hundreds of emails received at a computer to deploy new forms of worms onto a victim's computer, similar to the "Morris Worm," these new email worms are designed to steal information or cripple the victim's computers. "Fig. 1" is an illustration of the classic email worm attack. Creating an email worm that will automatically spread itself within a confined environment via email is crucial in modern day cybersecurity. Hackers are continually improving their methods of creating malwares, therefore it is very important to stay informed about the new trends and test that the systems that we are defending or monitoring are up to date to the latest methods of attacking. The project will show that although modern email services such as gmail, hotmail or yahoo Mail have built very tight defence mechanics to prevent attackers from spoofing or hijacking an email session an attacker may still infect a victim's computer using emails but can use a backdoor approach to steal the victim's contacts and send them to the attacker which can send the worm the new found contacts, hence replicating itself.

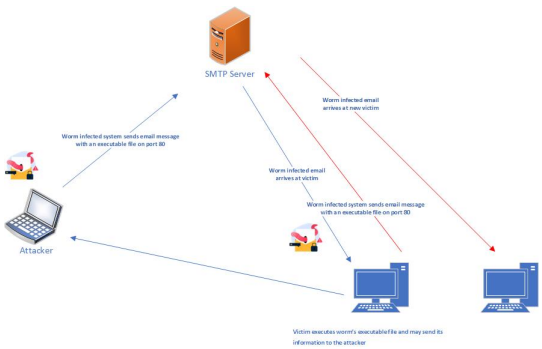

Fig. 1. Diagram of a classic worm attack.

Creating an email worm that will automatically spread itself within a confined environment via email is crucial in modern day cybersecurity. Hackers are continually improving their methods of creating malwares, therefore it is very important to stay informed about the new trends and test that the systems that we are defending or monitoring are up to date to the latest methods of attacking. The project will show that although modern email services such as gmail, hotmail or yahoo Mail have built very tight defence mechanics to prevent attackers from spoofing or hijacking an email session an attacker may

still infect a victim's computer using emails but can use a backdoor approach to steal the victim's contacts and send them to the attacker which can send the worm the new found contacts, hence replicating itself.

II. Literature Review and Existing Approach

Email worms continue to pose significant cybersecurity threats, evolving alongside advancements in technology. These malicious programs exploit email as a propagation medium to disrupt systems, compromise sensitive data, and cause widespread damage. Early examples such as Melissa, Sobig, and Mydoom demonstrated the devastating impact of such attacks. Notably, Mydoom remains infamous for being the fastest-spreading email worm in history, crippling networks worldwide. This review explores the taxonomy, detection methods, mitigation strategies, and challenges in combating email worms, alongside potential future directions for research.

Taxonomy of Email Worms

Understanding email worms necessitates categorizing them based on their behavior and propagation mechanisms:

Mass-Mailing Worms

Mass-mailing worms primarily spread by leveraging victims' email contact lists to disseminate malicious emails. These worms often disguise themselves with familiar sender details to increase the likelihood of being opened.

File-Infecting Worms

These worms propagate by embedding malicious code into executable files. When users open these files, the worms infect the host system and spread to other vulnerable systems.

Hybrid Worms

Hybrid worms combine multiple propagation methods, including emails, instant messaging, and network exploits. Their adaptability makes them particularly challenging to detect and contain.

This taxonomy provides a foundational understanding of email worm behavior and facilitates targeted defense strategies.

Advances in Worm Detection

Modern detection methods have evolved to address the increasing sophistication of email worms:

Behavioral Analysis

Behavioral analysis identifies anomalies in email metadata, such as unusually high volumes of sent emails or abnormal attachment patterns, which are indicative of worm activity.

Artificial Intelligence and Machine Learning

AI and machine learning algorithms analyze vast datasets to identify patterns characteristic of worm activity. These methods enhance detection accuracy by adapting to new threats.

Anomaly-Based Intrusion Detection Systems (IDS)

Anomaly-based IDS monitors network traffic for deviations from normal behavior, enabling real-time detection of worm outbreaks at the server level.

These methods represent significant improvements over traditional heuristic approaches, which relied heavily on predefined signatures.

Challenges in Detection and Mitigation

Despite advancements, several challenges hinder effective email worm detection and mitigation:

Polymorphic and Metamorphic Worms

These worms modify their code with each iteration to evade signature-based detection systems.

Zero-Day Worms

Exploiting previously unknown vulnerabilities, zero-day worms remain undetectable until the vulnerability is identified and patched.

Social Engineering

Many worms rely on phishing tactics to trick users into opening malicious attachments or links, highlighting the importance of addressing human error.

These challenges underscore the need for adaptive and robust detection systems.

Existing Mitigation Strategies

Several strategies have been employed to mitigate the impact of email worms:

Email Filtering

Advanced email filtering tools scan incoming messages for malicious attachments and suspicious links, reducing the risk of worm propagation.

User Awareness Campaigns

Educating users on recognizing phishing emails and safe email practices is critical in reducing the likelihood of human error.

Patch Management

Timely updates to software and operating systems address known vulnerabilities, preventing worms from exploiting them.

While effective, these strategies are not foolproof and require continuous enhancement to keep pace with evolving threats.

Research Gaps and Future Directions

Several limitations in current approaches present opportunities for further research:

Real-Time Detection for Large-Scale Traffic

Existing systems often struggle to analyze high volumes of email traffic in real time without sacrificing accuracy.

Technology Category	Technology	Selection Criteria
Infrastructure	Virtual Machines	The CLaaS environment is great place for the team to collaborate and test
Infrastructure	Linux OS	The Linux operating system is simple to use and compatible with the other tools we needed to use
Infrastructure	Windows OS	The Windows operating system is simple to use and present on the local devices used for the project
Code	Python	All teammates have python skills and it is an effective language for the functions of the Email Worm
Email	Gmail	Gmail as an email provider is easy-to-use and set up
Email	Thunderbird	By installing an email client on the virtual machine, access to contacts is easier
Email	IMAP and POP3	The Post Office Protocol downloads the email from the email provider to the client locally on the VM. This makes programmatic access much easier.

TABLE I
TECHNOLOGIES USED FOR CREATING THE EMAIL WORM

Collaborative Threat Intelligence

Increased collaboration among organizations to share threat intelligence could improve detection and mitigation efforts.

Balancing Accuracy and Scalability

Developing systems that maintain high detection accuracy while scaling to enterprise-level traffic volumes is an ongoing challenge.

Future research should focus on adaptive, scalable solutions that address these gaps and enhance the resilience of email systems against worm attacks.

III. PROJECT ARCHITECTURE

The Email Worm is designed to be delivered as an email attachment and designed to be executed on a victim's machine. It will propagate or spread by itself to new victims that it acquires from the current victim's contact list. The Email Worm will run indefinitely or until it no longer can find new email addresses from the computer it is running on.

The technologies used for this project are listed by technology type in the following table.

In our recent approach, an email worm contains malicious code, and is capable of performing the following tasks once sent to the victim and opened as mentioned above in "Fig. 2" by (X). The attachment (often an executable file) contains

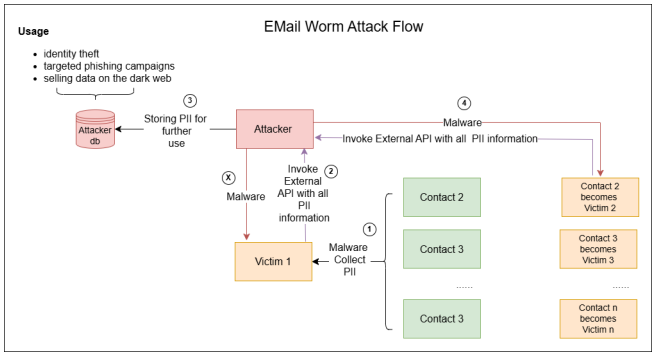

Fig. 2. Architecture of the implemented Email Worm.

malicious code. When opened or executed it performs the following:

1) **Reading Contacts and Info**
   The decision was made to parse email addresses from the locally installed email client. Accessing the victim's list of contacts directly from the email provider but they often block programmatic access or require a PIN.
   - Determines OS (Windows or Linux) of victims device to discover location of email client installation
   - Accesses local contact lists stored in email clients (e.g., Thunderbird) or address books on the victim's device
   - Discovers smtp server and port that is used in email client profile / configuration file
   - The code might also scan files and folders for email addresses, phone numbers, or other contact information
2) **Calling APIs**
   - Connects to external APIs or servers controlled by the attacker
   - Uploads the collected contact information to the attacker's server
   - Fetches additional instructions or malicious payloads to execute further attacks
3) **Collecting Information**
   The attacker can use the uploaded data (contact information) to:
   - Send spam or phishing emails: Exploiting the collected contacts to spread the worm further
   - Build a database: For other malicious activities like identity theft, targeted phishing campaigns, or selling data on the dark web
4) **Sending Again**
   This process will create the email and get the recipient's address from the attacker's database. The email delivers the malicious payload via an attachment. The construction of the email must be in a way where it tricks the victim to open the attachment. After collecting contacts, the worm can:
   - Use the victim's email account or spoof their identity to send emails to the collected contacts, making

Fig. 3. Test for finding contacts onto the victim

Fig. 4. API catching the email contact and sending email work to new found contacts

the emails appear more trustworthy
- Include itself as an attachment or malicious link in these emails, propagating further

Install Malicious File: A malicious file could be installed on the victim's computer. This could be a rootkit for use in the future. Other capabilities in this file could range from a Software Keylogger to a script that looks for data to exfiltrate and send back to the attacker. For this project, we deemed this "out of scope" and focused on demonstrating the Email Worm's capability.

IV. Experiments and Results

Experiments were designed to align to the 4 main capabilities detailed in the Email Worm Attack Flow in the Project Architecture section. Several challenges arose as a result of modern email and computing security. Mitigations had to be designed and in some cases caused architectural modifications.

- **Email contact search:** Search the victims contacts for email addresses to be used for expanding the attack.
  - First, test locally by downloading Thunderbird to a laptop and create email contacts.
  - Search installation for profile path. This is where contacts can be extracted as well as other information like SMTP server settings.
  - Next, move the script to the CLaaS VMs and modify code to work on Linux as well as Windows.

Challenges: The victim's device could have a different operating system which may install the profile and other files to different locations. The code searches for the installation path and locates the prefs.js file. Testing ensured the path was properly created regardless of where Thunderbird is installed. Tests passed for Windows 11 and Linux Ubuntu 20.04.5 Operating Systems.

- **Worm Propagation Simulation:** Simulate the behavior of a typical email worm in a controlled virtual environment, sending the worm through various email systems and analyzing the speed and methods of propagation.
  - Connectivity test to attacker API using POST.
  - JSON format of collected contacts email addresses. While the connection to the attacker VM hod to occur on the CLaaS environment, the JSON formatting was tested locally as the python Integrated Development Environment (IDE) was more stable and provided features to expedite troubleshooting.
  - Programmatically send an email to received email address/es and attach the malicious payload.

Challenges: There are considerable security controls that have been implemented by Google to protect their GMail product and its customers. Many oscillations with trial by error occurred to navigate these controls. GMail often blocked anything it deemed malicious including python files, executables, and even zipped payloads. Since we used python, we need a way to create an executable that could stand alone (no guaranteed the victim has installed python). Through research and many tests we discovered that Nuitka is a free Python Compiler and created a .bin file. The .bin file format is currently allowed by GMail's anti-malware scans.

- **Malicious code execution:** Simulate the victim's actions upon receiving the email to discover "usability" and minimize the tasks required to execute the malicious file.
  - Packaging must entice the victim to open the file.
  - File will not have permissions so a "desktop" launcher must accompany the payload.
  - Launcher must dynamically point ot he executable file.

Challenges: Operating Systems have security controls to make execution of code difficult by prompting the user as well as setting default permissions for file to not allow execution. These mandatory controls were navigated with a launcher file and while it prompts the user for permission, we felt the flow was smooth and the user would grant permission at a high enough percentage for a successful propagation.

- **Performance and Impact Analysis:** Analyze how different mitigation techniques impact network performance and the containment of the worm attack.
  - Examine user experience.
  - Flag tedious tasks for analysis and new design.
  - Put all actions and tasks end-to-end to evaluate the complete attack.

  Challenges: Users have a short attention span and expect quick responses to actions taken on a computer. We had to redesign our attack many times to eliminate tasks we deemed too time consuming to reasonably expect a user to complete them.

V. Summary and Conclusion

In summary, this project aims to address the ongoing challenge of email worm attacks by analyzing their propagation mechanisms and evaluating current detection strategies. The experiment provided valuable insights into how well modern detection technologies, including email vendors and operating systems, can identify and contain email worms before they cause significant damage. The project also explored potential improvements in existing systems, contributing to the development of more robust and effective email security solutions.

Through this research, the goal was to better understand the threat of email worms and propose new, effective mitigation strategies that enhance cybersecurity defenses.

****Interesting development during project: We discovered many external IP addresses attempting to connect to our API while it was running. Further analysis is needed to understand how our API was discovered and potential "honeypot" development to learn more about these potential attacks and the behaviors of the suspected adversaries.*

REFERENCES

[1] Palmer, D. (2019, July 26). MyDoom: The 15-year-old malware that's still being used in phishing attacks in 2019. *ZDNet*. Retrieved from https://www.zdnet.com/article/mydoom-the-15-year-old-malware-thats-still-being-used-in-phishing-attacks-in-2019/

[2] Saini, N., Pandey, N., & Singh, A. P. (2016, December). Analyzing and developing security techniques for worms in cognitive networks. *IEEE Xplore*. https://doi.org/10.1109/WCNCW.2016.7919554

[3] Security.org. (n.d.). Computer worm. Retrieved from https://www.security.org/antivirus/computer-worm/

[4] Park, I., Sharman, R., Rao, H. R., & Upadhyaya, S. (2007). Short-term and total life impact analysis of email worms in computer systems. *Decision Support Systems*, 43(3), 827–841. https://doi.org/10.1016/j.dss.2006.12.014

[5] Li, P., Salour, M., & Su, X. (2008). A survey of Internet worm detection and containment. *IEEE Communications Surveys & Tutorials*, 10(1), 20–35. https://doi.org/10.1109/COMST.2008.4483668

[6] Obimbo, C., Speller, A., Myers, K., Burke, A., & Blatz, M. (2018). Internet worms and the weakest link: Human error. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. https://doi.org/10.1109/CSCI.2018.8947674

[7] FBI. (2018). Morris Worm: 30 years since the first major attack on the Internet. Retrieved from https://www.fbi.gov/news/stories/morris-worm-30-years-since-first-major-attack-on-internet-110218

[8] FBI. (n.d.). Business email compromise. Retrieved from https://www.fbi.gov/how-we-can-help-you/scams-and-safety/common-frauds-and-scams/business-email-compromise

[9] Insights Desk. (2023, November 2). Computer worms: A threat to beware. Retrieved from https://www.itsecuritydemand.com/insights/security/computer-worms-a-threat-to-beware/

[10] Kaspersky Encyclopedia. (n.d.). Email worms. Retrieved from https://encyclopedia.kaspersky.com/knowledge/email-worm/