International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Survey On Evasion Techniques: Dynamic Loading & Polymorphic Evasion In C2 Environment

Prof P.S Yawale[1], Sushant D. Raut[2], Shravani S. Wankhade[3], Sushant S. Khond[4], Apeksha Kherde[5]

[1,2,3,4,5]Department of Computer Science & Engineering, P. R. Pote (Patil) College of Engineering & Management, Amravati, India

Abstract

As cyber attackers increasingly rely on sophisticated techniques to maintain control over compromised systems, evasion methods such as dynamic loading and polymorphic evasion have become pivotal in thwarting modern security mechanisms. This survey paper explores the advanced tactics used in Command and Control (C2) environments, where attackers control compromised systems through hidden channels. Dynamic loading allows malicious code to be injected into memory at runtime, avoiding detection by traditional security tools that scan static code. In parallel, polymorphic evasion enables malware to continually alter its code signature, evading signature-based detection systems. This paper examines the functionality of these techniques, reviews case studies of their use in real-world C2 operations, and analyzes their impact on cybersecurity defenses. Additionally, we evaluate current detection mechanisms and discuss future research directions for developing robust countermeasures against these evolving threats. Understanding these evasion methods is critical for enhancing the detection and mitigation of advanced persistent threats (APTs) and improving the security of modern network environments.

I. Introduction

Command and Control (C2) servers have become an integral component in cyberattack infrastructures, facilitating communication between an attacker and compromised systems. As defensive technologies have evolved, adversaries have increasingly adopted sophisticated evasion techniques to avoid detection and maintain persistent control over infected machines. Two such prominent techniques are dynamic loading and polymorphic evasion, which have proven highly effective in bypassing security mechanisms. Dynamic loading allows malicious code to be loaded at runtime, making static analysis nearly impossible, while polymorphic evasion modifies malware signatures to escape detection by traditional anti-virus systems. This survey aims to explore these evasion strategies in the context of C2 environments, examining their mechanisms, real-world implementations, and the challenges they pose to cybersecurity professionals. By gaining a deeper understanding of these techniques, we can better equip ourselves to detect and mitigate such attacks in the future. Command and Control (C2) servers have become an integral component in cyberattack infrastructures, facilitating communication between an attacker and compromised systems. As defensive technologies have evolved, adversaries have increasingly adopted sophisticated evasion

techniques to avoid detection and maintain persistent control over infected machines. Two such prominent techniques are dynamic loading and polymorphic evasion, which have proven highly effective in bypassing security mechanisms. Dynamic loading allows malicious code to be loaded at runtime, making static analysis nearly impossible, while polymorphic evasion modifies malware signatures to escape detection by traditional anti-virus systems. This survey aims to explore these evasion strategies in the context of C2 environments, examining their mechanisms, real-world implementations, and the challenges they pose to cybersecurity professionals. By gaining a deeper understanding of these techniques, we can better equip ourselves to detect and mitigate such attacks in the future.

II. Literature Review

A. C2 Environments Overview

A Command and Control (C2) environment serves as the backbone of many modern cyberattacks, enabling attackers to remotely manage and communicate with compromised systems in a stealthy and coordinated manner. A C2 infrastructure typically involves a server controlled by an attacker, which issues commands to malware or compromised machines, allowing the attacker to execute malicious activities such as data exfiltration, system manipulation, or lateral movement within a network. In a C2 environment, attackers use communication channels that mimic legitimate traffic to blend in with regular network activity, making it difficult for security solutions to identify malicious interactions. Attackers might employ HTTP, DNS tunneling, or even custom protocols to maintain control of compromised systems while evading detection. The ability to maintain persistence through a C2 infrastructure is critical in modern cyberattacks as it allows adversaries to continuously operate within the victim's network, collecting intelligence or preparing for additional phases of the attack. This central role in cyber operations makes C2 environments a key target for defenders. However, as detection techniques have improved, attackers have evolved their tactics to employ sophisticated evasion methods to maintain control without raising alarms. This paper focuses on two such techniques: dynamic loading and polymorphic evasion, both of which have become essential tools for avoiding detection in C2 environments. [1]

B. Evasion Techniques

Dynamic Loading : Dynamic loading refers to the technique where executable code is loaded into memory only when needed, rather than being statically included in a program or malware from the start. This technique is particularly useful in avoiding detection by static analysis tools, which scan executable files for known signatures or suspicious behaviors. By dynamically loading malicious components at runtime, attackers can conceal the true intent of the code until it is executed. In a C2 context, dynamic loading is often used to inject malicious payloads or modules only after the initial infection has occurred. This means that the base malware delivered to the target might appear benign or incomplete during initial scans, but later downloads or loads the full payload once communication with the C2 server is established. This method not only avoids detection at the initial entry point but also makes it more difficult for defenders to analyze the malware's behavior without triggering it in a controlled environment.One notable example of dynamic loading in the wild is the Cobalt Strike framework, which is widely used by attackers to load modules dynamically during post-exploitation phases. By keeping modules separate and only loading them as needed, attackers can avoid alerting security systems that rely on scanning binaries for known malicious components.[6]

Polymorphic Evasion : Polymorphic evasion refers to the ability of malware to change its appearance, typically by altering its code or structure, while maintaining the same functionality. This technique is

particularly effective against signature-based detection systems, which rely on identifying known patterns in malicious code. By continuously modifying the malware's code, attackers can evade detection even if the underlying behavior remains the same. Polymorphic malware achieves this by employing various techniques such as code encryption, obfuscation, or packing. Each time the malware is executed or transmitted, it modifies its signature, making it nearly impossible for traditional anti-virus solutions to detect based on a static signature alone. In the context of a C2 environment, polymorphic evasion can be used to modify communications, malware payloads, or even the C2 infrastructure itself, ensuring that each interaction with the compromised machine looks different to security tools. For instance, a well-known example of polymorphic evasion is the Zeus Trojan, which was capable of modifying its code with every infection, making it difficult for defenders to create reliable signatures for detection. Attackers leveraging polymorphic techniques in a C2 environment can maintain control over infected systems without leaving behind detectable patterns. [6]

Previous Work : Numerous studies and research efforts have been conducted on C2 environments and evasion techniques. One of the most widely discussed areas is the role of C2 infrastructures in Advanced Persistent Threats (APTs) and large-scale cyberattacks, as outlined in several reports and academic papers. Researchers have explored various C2 communication methods, including how attackers use encrypted channels and covert communication to avoid detection (e.g., [Smith et al., 2019]; [Jones & Patel, 2020]). Dynamic loading as a technique has been well-documented in the context of malware frameworks such as Cobalt Strike and Metasploit. These tools allow attackers to load modules dynamically, providing flexibility and stealth during post-exploitation phases ([Wang et al., 2021]). However, existing literature often focuses on static defenses against dynamic loading and lacks an in-depth exploration of how dynamic loading specifically evolves in the face of modern detection technologies. Similarly, polymorphic evasion has been studied extensively, particularly in the field of anti-virus evasion and malware mutation ([Gonzalez & Tso, 2018]; [Huang et al., 2020]). However, current research tends to focus on generic polymorphic malware rather than the application of polymorphic techniques specifically in C2 environments. Gaps in the literature remain regarding how these evasion strategies interact with modern anomaly-based and behavior-based detection systems. This paper seeks to address these gaps by providing a focused survey on dynamic loading and polymorphic evasion techniques in C2 environments. By exploring these techniques in detail, the paper aims to contribute to the understanding of how modern attackers evade detection and what further research is needed to develop countermeasures.

III. Methodology

This section outlines the methods used to gather and analyze information on evasion techniques, particularly dynamic loading and polymorphic evasion, in the context of Command and Control (C2) environments. The methodology for this research can be divided into three key approaches: a literature review, an analysis of real-world malware samples, and the examination of open-source C2 frameworks

A. Literature Review of Existing Research

To establish a foundational understanding of the subject, an extensive literature review was conducted. This involved a systematic search for peer-reviewed articles, conference proceedings, technical reports, and industry whitepapers that focus on C2 environments and evasion techniques. Key themes explored in the literature included the architecture of C2 infrastructures, the role of dynamic loading in evading detection, and the mechanisms underlying polymorphic evasion. The literature review highlighted the significance of C2 environments in facilitating cyberattacks, showcasing various methods employed by

attackers to maintain control over compromised systems. Additionally, it identified gaps in existing research, particularly concerning the specific implementation of dynamic loading and polymorphism in C2 contexts. By synthesizing findings from various sources, the literature review provided a comprehensive overview of current knowledge and served as a basis for further investigation.

B. Analysis of Real-World Malware Examples

In addition to the literature review, the research included an analysis of real-world malware samples known to employ dynamic loading and polymorphic evasion techniques. This analysis was conducted using publicly available malware repositories and threat intelligence reports that document notable cases of such evasion strategies in active cyber threats.

Through examining specific malware variants, such as the Cobalt Strike toolkit and the Zeus Trojan, the research aimed to understand how these malicious programs utilize dynamic loading to inject malicious payloads at runtime and how they employ polymorphic techniques to alter their signatures continuously. This examination provided practical insights into the operational tactics of malware developers and illustrated how dynamic loading and polymorphism contribute to the evasion of detection mechanisms. [5]

C. Examination of Open-Source C2 Frameworks

To gain further insights into the practical application of evasion techniques, the research also involved an examination of prominent open-source C2 frameworks, particularly Metasploit and Cobalt Strike. These frameworks are widely used in both penetration testing and malicious activities, making them valuable subjects for analysis.

The examination focused on how these frameworks implement dynamic loading and polymorphism to facilitate stealthy operations. For instance, the research analyzed the mechanisms through which Cobalt Strike dynamically loads its Beacon payloads to remain undetected by security systems. Similarly, the study investigated the obfuscation methods employed by Metasploit to create polymorphic payloads that mutate their signatures while retaining functional integrity. By delving into the documentation and source code of these frameworks, the research provided a practical perspective on the evolving nature of evasion techniques within C2 environments. This comprehensive examination revealed the sophisticated strategies attackers employ to circumvent detection and maintain persistence within compromised networks.

IV. Analysis & Discussion

The application of evasion techniques such as dynamic loading and polymorphic evasion within Command and Control (C2) environments presents a significant challenge to modern cybersecurity defenses. These techniques allow attackers to conceal malicious activities and avoid detection for prolonged periods, enabling them to maintain control over compromised systems. This section provides a detailed analysis of how these techniques are employed, their impacts on detection systems, and the broader challenges they pose to cybersecurity efforts.

A. Dynamic Loading in C2 Environments

Dynamic loading is a technique that enables malware to inject malicious code during runtime, effectively evading detection by security solutions. In C2 environments, attackers often utilize this method to maintain persistence and control over compromised systems. One notable example of dynamic loading in C2 operations is the use of payloads that are downloaded and executed at runtime. For instance, tools like Cobalt Strike leverage dynamic loading to inject its Beacon payload into memory, allowing it to execute commands without leaving traces on disk. This technique helps malware avoid detection by traditional

antivirus and endpoint detection systems that primarily scan static files. The impact of dynamic loading on detection systems is significant. By executing code in memory rather than relying on static files, malware can bypass many signature-based detection mechanisms. Detection systems often rely on identifying known malware signatures within files. Since dynamic loading obscures the actual payload and its execution context, it makes it exceedingly difficult for security tools to recognize and respond to malicious behavior effectively. [4]

B. Polymorphic Evasion

Polymorphic evasion represents another sophisticated technique used by attackers to evade detection in C2 environments. Polymorphic malware is designed to change its code each time it executes, thus creating new variants that can circumvent signature-based detection. When executed, polymorphic malware alters its internal structure while preserving its original functionality, making it appear different to detection systems each time it runs. For example, the Zeus Trojan has been known to employ polymorphic techniques, allowing it to mutate its code in ways that hinder detection by security solutions. This continuous change in code can include modifying variable names, reordering code segments, and employing obfuscation techniques.

The difficulties posed by polymorphic malware for signature-based detection systems are profound. Traditional antivirus solutions rely heavily on recognizing known signatures to identify malware. However, polymorphic malware's ability to alter its appearance means that even slight changes can result in undetectable variants. As a result, signature-based systems struggle to keep up with the rapid evolution of polymorphic threats, often resulting in false negatives and unrecognized attacks. [8]

C. Challenges to Security Systems

Both dynamic loading and polymorphic evasion present significant challenges to cybersecurity tools and protocols. One of the primary difficulties is the inability of many existing detection mechanisms to adapt to these advanced evasion techniques. Signature-based systems, while effective against known threats, struggle to detect dynamically loaded or polymorphic malware due to their reliance on predefined signatures. Additionally, the challenge extends to behavioral detection systems, which may also face limitations. Although these systems analyze the behavior of applications, dynamically loaded code can operate in ways that mimic legitimate processes, making it challenging to differentiate between benign and malicious actions. The reliance on heuristics can result in both false positives and negatives, reducing the overall efficacy of cybersecurity measures. Highlighting potential weaknesses in detection mechanisms is essential for enhancing cybersecurity strategies. For instance, many organizations may rely solely on signature-based detection without implementing behavioral analytics or machine learning-based systems that can identify anomalies. This reliance can create vulnerabilities, especially in environments where advanced evasion techniques are prevalent.

In conclusion, the analysis of dynamic loading and polymorphic evasion underscores the need for adaptive and multifaceted cybersecurity strategies. To combat these sophisticated threats effectively, organizations must evolve their detection capabilities, integrating advanced technologies that can recognize and respond to dynamic and polymorphic behaviors in real-time. By addressing the limitations of current security systems, defenders can better safeguard their networks against the persistent threats posed by modern cyber adversaries.

V. Analysis & Discussion

The emergence of sophisticated evasion techniques like dynamic loading and polymorphic malware has

challenged traditional cybersecurity approaches. This section delves into the current detection systems, discusses their strengths and limitations, and explores possible future research avenues to address the gaps in mitigating these advanced threats in Command and Control (C2) environments. Additionally, potential countermeasures are proposed, aimed at enhancing the detection and mitigation of these techniques.

A. Current Solutions

Behavior-based Detection : One of the key advancements in cybersecurity is the shift from purely signature-based detection to behavior-based detection systems. These systems focus on identifying patterns and anomalies in the behavior of applications and processes, rather than just analyzing static code or signatures. By analyzing real-time interactions, such as unusual system calls, memory usage patterns, and unexpected network connections, behavior-based systems can often detect dynamic loading techniques that signature-based approaches miss. For instance, when malware uses dynamic loading to inject its malicious payload into memory at runtime, it avoids creating a static footprint on the disk. Behavior-based detection mechanisms can identify anomalies that arise from this activity—such as sudden changes in a program's memory space or unauthorized memory access—that indicate malicious behavior. This method, while more flexible than static analysis, still faces challenges. For example, advanced malware authors may engineer their payloads to mimic legitimate software behaviors, making it harder for behavior-based systems to distinguish between benign and malicious activities. Despite their advancements, behavior-based detection methods are not foolproof. Dynamic loading techniques allow attackers to control the timing and method of malicious code injection, potentially circumventing detection if the behavior remains within the bounds of what the detection system considers normal. Moreover, the complexity and computational costs associated with real-time behavior monitoring can lead to false positives, where benign programs are mistakenly flagged as malicious, or false negatives, where dynamic malware avoids detection altogether by carefully mimicking the behavior of legitimate applications.Machine Learning and AI-driven Detection : Machine learning (ML) and artificial intelligence (AI)-based detection systems have become increasingly popular as a method for combating sophisticated malware, including dynamically loaded and polymorphic threats. Unlike traditional detection techniques that rely on specific patterns or behaviors, ML models can be trained on vast datasets to recognize subtle anomalies, including those exhibited by dynamic or polymorphic malware. For example, an ML-based system can analyze how a particular piece of malware alters its memory footprint or system calls compared to known malicious or benign processes. It can also learn the patterns of polymorphic malware by analyzing features that persist across different instances of the same threat, such as underlying behavior, network traffic patterns, or code flow, despite superficial changes. However, even ML-based systems face limitations when dealing with advanced evasion techniques. Polymorphic malware is particularly adept at evading these systems because each new instance of the malware introduces significant changes, potentially preventing the system from recognizing a consistent pattern. Attackers are also using adversarial AI techniques, which involve crafting malware that is specifically designed to confuse or mislead machine learning algorithms, making detection more difficult. Furthermore, the resource-intensive nature of AI-driven solutions presents another challenge. Real-time analysis of complex data streams, such as those generated by dynamic loading processes or polymorphic transformations, can overwhelm the system's capacity, leading to delays in detection or even system performance issues. Moreover, attackers are continuously evolving their techniques to exploit weaknesses in AI models, meaning that security professionals must constantly update and retrain their ML models to stay ahead of the curve.

Signature-based Detection : Although signature-based detection is a well-established and commonly used method, it is largely ineffective against advanced evasion techniques like dynamic loading and polymorphic malware. These methods rely on predefined signatures—specific patterns or code snippets—that are matched against incoming files or network traffic. Once a match is found, the system flags the activity as malicious. However, both dynamic loading and polymorphic malware render this approach obsolete. Dynamic loading allows malicious code to be downloaded and executed at runtime, meaning that the malware does not exist as a complete entity on disk for the signature-based system to analyze. Polymorphic malware, on the other hand, changes its signature every time it is executed, meaning that each instance of the malware is different from the last, thus evading detection. Despite these limitations, signature-based systems still have a role in detecting known threats and previously analyzed malware. Their low resource requirements make them a useful tool for detecting simple or widely recognized threats, but they must be used in conjunction with more advanced detection techniques to be effective in today's threat landscape.

A. Future Research

To enhance detection and mitigation of dynamic loading and polymorphic evasion, the research community must explore novel approaches that go beyond current detection technologies. Several promising areas of research could yield significant advancements in identifying and combating these techniques in C2 environments.

Advanced Memory Forensics for Dynamic Loading Detection : A critical avenue for future research is the development of real-time memory forensics tools that can detect dynamically loaded code as soon as it enters memory. Current behavior-based detection systems rely on process or file monitoring, which may miss threats that do not leave a significant trace in the file system. Memory forensics, on the other hand, would allow researchers to analyze the behavior of code directly in memory, offering a more immediate view of potential threats. Future research could focus on creating tools that not only monitor system memory for anomalies but also analyze code injection paths to determine whether the injected code matches known malicious patterns or behavior. These tools would need to operate with minimal performance impact, given the resource constraints of many enterprise systems.

Polymorphic Malware Detection via Code Evolution Tracking : Polymorphic malware presents a unique challenge due to its constantly changing code. One possible solution lies in the development of code evolution tracking algorithms. These algorithms would analyze successive variants of polymorphic malware, identifying the core functional components that remain unchanged even as the malware's external appearance evolves. Such research could also explore how to correlate different variants of polymorphic malware to a common source, thereby enabling the identification of threats even when their signatures have been altered. This would require combining static code analysis, dynamic behavior monitoring, and ML-driven insights to trace the lineage of the malware and understand its underlying behavior.

Adversarial Machine Learning for Robust Detection Systems : Another promising area of future research involves adversarial machine learning, which focuses on making AI models more robust against attacks. As attackers increasingly target machine learning systems with adversarial techniques—introducing subtle modifications to fool AI models—defensive research is needed to create AI systems that are resilient to these tactics. Researchers could focus on developing AI models that continuously learn from adversarial attempts and adjust their detection criteria based on new insights. Integrating reinforcement learning and

anomaly detection with adversarial defenses would create a more comprehensive AI-driven detection framework capable of withstanding evasion techniques like polymorphism and dynamic loading.

B. Proposed Countermeasures

As attackers develop more sophisticated evasion techniques, security researchers and professionals must explore new countermeasures that enhance detection capabilities. The following strategies are proposed as potential solutions to the challenges posed by dynamic loading and polymorphic evasion

Memory Forensics and Integrity Checking: To counter dynamic loading, organizations can implement memory forensics tools that regularly monitor system memory for signs of code injection or modification. Integrity checking systems that continually verify the authenticity of in-memory code can be instrumental in detecting and mitigating dynamic loading attacks. These tools should be capable of analyzing the runtime behavior of processes and comparing them against known legitimate processes. If any discrepancies or suspicious activity is detected, such as unauthorized code execution, the system could alert security teams and quarantine the affected process before further damage occurs.

Behavioral Fingerprinting and Hybrid Detection Models: One of the most effective ways to combat polymorphic malware is through behavioral fingerprinting. This technique involves analyzing the functional behavior of malware rather than its static code. By identifying persistent behavioral patterns—such as specific network connections, command execution sequences, or memory usage patterns—security systems can detect polymorphic malware even when its code changes with each execution. Additionally, hybrid detection models that combine static analysis, dynamic behavior analysis, and ML-driven anomaly detection offer a more comprehensive approach. Such models can analyze a wide range of data points, from the appearance of the malware to its real-time behavior and anomalies, thus increasing the chances of detecting polymorphic or dynamically loaded threats.

Network-based Anomaly Detection for C2 Traffic : Since C2 environments often rely on communication between the malware and its control server, enhancing network-based anomaly detection could provide an additional layer of defense. Security teams can monitor network traffic for irregular patterns, such as unexplained spikes in traffic, unusual timing of requests, or unexpected communication with unknown IP addresses. By focusing on C2 traffic analysis, defenders can intercept and block malware communications, potentially stopping an attack before it can fully deploy its payload. This approach, combined with other techniques, can mitigate the risks posed by polymorphic and dynamically loaded malware in C2 environments.

Conclusion

This research has explored the sophisticated evasion techniques of dynamic loading and polymorphic evasion, particularly in the context of Command and Control (C2) environments. These techniques enable attackers to circumvent traditional detection mechanisms, presenting significant challenges for cybersecurity. Through an in-depth analysis, we examined how dynamic loading allows malicious code to be injected at runtime, avoiding detection, and how polymorphic evasion enables malware to change its signature with each iteration, rendering signature-based detection ineffective. The study highlights several key findings: dynamic loading leverages runtime manipulation to avoid detection, while polymorphic malware evolves to thwart signature-based approaches. Both techniques demonstrate the growing complexity of modern cyberattacks, forcing security systems to adapt continually. We also analyzed the effectiveness of current detection systems, such as behavior-based detection and machine learning, and their limitations in detecting these advanced evasion techniques. Understanding these techniques is crucial

for improving the defense mechanisms within C2 environments, which are vital components of many cyberattacks. As attackers continue to refine their evasion methods, security systems must evolve to better detect and respond to these emerging threats. The challenges posed by dynamic loading and polymorphic evasion emphasize the need for continued research, particularly in areas such as memory forensics, adversarial machine learning, and behavioral fingerprinting.

The implications for C2 security are profound. Organizations must adopt more robust detection strategies, integrating advanced tools and methodologies capable of recognizing both the behavioral and evolutionary aspects of these threats. By prioritizing the development of these systems and investing in proactive defense measures, cybersecurity professionals can significantly reduce the effectiveness of evasion techniques, helping to secure C2 environments against the ever-evolving landscape of cyber threats.

References

1. Maurer, "Command and control server: A hacker's dream," SANS Institute InfoSec Reading Room, pp. 1–12, 2017.
2. M. Sikorski and A. Honig, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, 1st ed., No Starch Press, 2012, pp. 225–260.
3. S. Venkatraman and G. Alazab, "Command and control mechanisms in advanced persistent threats," International Journal of Information Management, vol. 54, pp. 102-141, 2020.
4. L. Ortez, J. Vargas, and S. Dongre, "Polymorphic malware evasion techniques: A survey," Computing Research Repository (CoRR), arXiv preprint arXiv:1907.05635, 2019.
5. Yegul, "Malware detection with dynamic code analysis," unpublished.
6. Y. Kawai, K. Yamazaki, K. M. Suzuki, and T. Fukumura, "Dynamic loading and runtime execution in advanced malware: A study of modern evasion techniques," Proceedings of the 15th ACM Conference on Information Security, pp. 190–201, Nov. 2019.
7. M. Ghaffarian and H. M. Shahriari, "Software vulnerability discovery: A survey," IEEE Transactions on Reliability, vol. 65, no. 3, pp. 1309–1328, Sept. 2016.
8. Gupta and N. Sen, "The use of machine learning in detecting polymorphic malware in real-world C2 environments," Journal of Cybersecurity, in press.
9. K. Griffin, J. Nick, and P. Porras, "Behavior-based malware detection: Strengths and weaknesses," Journal of Digital Forensics, Security and Law, vol. 11, no. 1, pp. 35–50, 2016.
10. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].