Trojanized “Claude Code”

GitHub Campaign & Vidar Stealer Distribution

CTI report

Erik Westhovens

06-04-2026

Cyber Threat Intelligence Report

Subject: Trojanized “Claude Code” GitHub Campaign & Vidar Stealer Distribution
Audience: SOC, Incident Response, Threat Hunting, Security Leadership
Date: April 2026

1. Executive Summary

In early April 2026, threat actors capitalized on the sudden exposure of development-related code associated with Anthropic’s “Claude Code.” What began as a moment of heightened interest within developer communities quickly evolved into a highly effective malware distribution campaign.

Within a remarkably short timeframe, attackers established a network of trojanized GitHub repositories presenting themselves as legitimate or “leaked” versions of the tool. These repositories were carefully designed to appear credible, often including detailed documentation, realistic project structures, and convincing naming conventions. For many users—particularly developers accustomed to exploring new tools and codebases—these signals were sufficient to establish trust.

The strength of this campaign lies not in technical novelty, but in its precise alignment with human behavior. There is no exploitation of software vulnerabilities, no phishing emails, and no reliance on complex intrusion techniques. Instead, the attack depends entirely on the user’s willingness to download and execute a file within a trusted environment.

Once executed, the malicious binary initiates a multi-stage infection chain centered around **Vidar Stealer**, supported by **GhostSocks proxy malware**. Together, these components enable attackers to extract sensitive data, hijack authenticated sessions, and repurpose compromised systems as part of a broader proxy network.

The implications are significant. By targeting developers and technically skilled users, the campaign increases the likelihood of accessing high-value assets such as source code repositories, API credentials, and cloud environments. At the same time, the use of session hijacking techniques allows attackers to bypass multi-factor authentication, reducing one of the most widely relied upon security controls.

From a defensive perspective, this campaign presents a challenge precisely because it does not resemble traditional threats. The initial activity—downloading code from GitHub and executing a binary—can appear entirely legitimate. As a result, detection opportunities are limited and often delayed until after sensitive data has already been exfiltrated.

This incident illustrates a broader shift in the threat landscape. Rather than focusing solely on vulnerabilities within systems, attackers are increasingly exploiting **trust in platforms, communities, and workflows**. In doing so, they are able to bypass many of the controls that organizations have spent years refining.

Any interaction with repositories claiming to provide access to “Claude Code,” particularly those offering leaked or enhanced versions, should therefore be treated as a potential precursor to compromise. Early recognition and rapid response are critical to limiting impact.

2. Threat Overview

2.1 Campaign Characterization

The campaign is best understood as a coordinated effort to distribute information-stealing malware through trusted developer channels. Its structure reflects a mature and repeatable approach rather than a one-off opportunistic attack.

At its core, the operation leverages **Vidar Stealer**, a well-established malware family known for its efficiency in extracting credentials and sensitive data. This is complemented by the deployment of **GhostSocks**, which extends the value of each infection by enabling proxy-based infrastructure abuse.

The combination of these components creates a dual-purpose infection model. On one hand, attackers gain immediate access to valuable data that can be used for account compromise or sold on underground markets. On the other, they establish longer-term access through proxy capabilities, allowing compromised systems to be used in future operations.

This layered approach reflects a clear focus on maximizing return on investment. Even in cases where credential theft yields limited results, the proxy functionality ensures that the infection remains monetizable.

2.2 Operational Model

The techniques observed in this campaign strongly suggest the use of a **Malware-as-a-Service (MaaS)** ecosystem. Rather than developing bespoke malware, threat actors rely on readily available tooling that can be deployed quickly and at scale.

This model enables a high degree of flexibility. Attackers can adapt their lures to align with current events—in this case, the exposure of Claude Code—while relying on proven malware for the underlying functionality. The result is a campaign that can be launched rapidly without sacrificing effectiveness.

The speed at which this operation emerged is particularly noteworthy. The transition from public awareness of the leak to active malware distribution occurred within a very short window. This suggests that the attackers were either closely monitoring developments in the AI ecosystem or were prepared to exploit such opportunities as they arose.

2.3 Targeting and Victim Profile

Unlike traditional campaigns that cast a wide net, this operation appears to focus on a more specific and valuable audience. Developers, security professionals, and AI engineers represent the primary targets, reflecting a deliberate strategy rather than random distribution.

These individuals occupy a unique position within organizations. They often have elevated access to systems, direct interaction with code repositories, and visibility into internal infrastructure. Compromising such users can therefore provide attackers with a foothold that extends far beyond the initial endpoint.

There is also a behavioral dimension to this targeting. Developers are accustomed to downloading, testing, and executing new tools, often with limited verification. This creates an environment in which malicious binaries can be introduced without triggering immediate suspicion.

The convergence of access and behavior makes this group particularly attractive to attackers. A single successful compromise can yield not only credentials but also insight into development pipelines, deployment processes, and internal architecture.

2.4 Objectives and Intent

The primary objective of the campaign is financial gain, but the methods used reflect a broader strategic approach. Rather than focusing on immediate disruption, such as ransomware deployment, the attackers prioritize stealth and long-term value extraction.

Credential harvesting is central to this strategy. By collecting login information and session data, attackers can gain access to a wide range of services without needing to trigger authentication mechanisms. This is particularly effective in environments where cloud-based platforms and single sign-on solutions are widely used.

Session hijacking further enhances this capability. By capturing authenticated sessions, attackers can bypass multi-factor authentication and operate as legitimate users. This reduces the likelihood of detection and allows for more subtle forms of exploitation.

The inclusion of proxy functionality introduces an additional layer of intent. By converting compromised systems into proxy nodes, attackers create a distributed infrastructure that can be used for future operations. This not only increases the value of each infection but also supports ongoing campaigns.

Taken together, these objectives reflect a shift toward **multi-layered monetization**, where each compromised system serves multiple purposes within the attacker’s ecosystem.

3. Campaign Context – The Role of the “Claude Code” Exposure

3.1 A Perfect Trigger Event

The exposure of code associated with Anthropic provided an ideal trigger for this campaign. The event generated immediate interest within technical communities, particularly among those working with AI tools and development frameworks.

This surge in attention created a unique opportunity. Users were actively searching for information, experimenting with new tools, and engaging in discussions about the leak. In this environment, the appearance of repositories claiming to provide access to the code was not only plausible but expected.

Attackers were quick to recognize this opportunity and act on it. By aligning their campaign with a real-world event, they were able to bypass one of the most important defenses in cybersecurity: skepticism.

3.2 Exploiting Trust in Platforms

GitHub plays a central role in modern software development. It is widely regarded as a trusted platform where developers collaborate, share code, and explore new ideas. This trust, while generally well-founded, can also be exploited.

By hosting malicious repositories on GitHub, attackers benefit from an implicit level of credibility. Users are less likely to question the legitimacy of a repository when it is presented within a familiar interface and accompanied by realistic documentation.

This is further reinforced by the structure of the repositories themselves. Many of the observed examples included elements designed to mimic legitimate projects, such as commit histories, file structures, and usage instructions. These details, while superficial, are often enough to convince users that the repository is genuine.

3.3 The Role of Discoverability

Visibility is a critical component of any malware distribution campaign, and in this case, attackers leveraged search engines and community channels to ensure that their repositories were easily discoverable.

By optimizing repository content for relevant keywords, they were able to appear in search results for queries related to Claude Code. This form of search engine manipulation increases the likelihood that users will encounter malicious content during normal browsing activity.

In addition to search engines, repositories were shared through forums, social media, and developer communities. This multi-channel approach amplifies reach and reinforces the perception that the repositories are widely known and accepted.

3.4 Psychological Drivers

At the heart of this campaign are two powerful psychological drivers: curiosity and urgency.

Curiosity motivates users to explore new tools and understand how they work. In the context of a high-profile code exposure, this curiosity is amplified by the desire to gain insight into cutting-edge technology.

Urgency, on the other hand, creates a sense of limited opportunity. Users may believe that access to the leaked code is temporary or that others are already taking advantage of it. This can lead to rushed decisions and reduced scrutiny.

By combining these factors, attackers create an environment in which users are more likely to take risks that they would otherwise avoid.

4. Attack Chain Assessment (Initial Stages)

4.1 Entry Through Normal Behavior

The initial stage of the attack is deceptively simple. The user searches for Claude Code, discovers a repository, and downloads the associated files. Each of these actions is entirely consistent with normal developer behavior.

This is what makes the campaign particularly challenging to detect. There is no clear boundary between legitimate and malicious activity at this stage. From the perspective of the system, the user is simply interacting with a trusted platform and executing a file.

4.2 Transition to Execution

The transition from download to execution marks the point at which the attack becomes active. The user, believing the file to be legitimate, runs the executable and initiates the infection chain.

Because this action is user-driven, it is not inherently suspicious from a technical standpoint. Security controls that rely on exploit detection or known malicious signatures may not be triggered, allowing the malware to execute without interference.

4.3 Early Indicators and Limitations

While early detection is possible, it requires a level of visibility and behavioral analysis that is not always present in enterprise environments. Indicators such as execution from download directories or unusual process behavior can provide clues, but they must be interpreted within context.

This highlights a key limitation in current defenses. Many organizations are well-equipped to detect known threats, but less prepared to identify attacks that operate within trusted workflows.

4.4 Preparing for the Payload Phase

Once execution has occurred, the system is effectively under the control of the dropper. The next phase involves the deployment of the core payloads, which will be explored in detail in the following section of this report.

At this point, the opportunity for early containment still exists, but it is rapidly diminishing. The longer the malware is allowed to operate, the greater the likelihood that sensitive data will be exfiltrated and the system will be integrated into the attacker’s infrastructure.

5. Malware Execution and Behavior Analysis

Once the victim executes the downloaded binary, the campaign transitions from social engineering into a technically mature and well-structured infection chain. What stands out is not extreme sophistication, but rather efficiency and reliability. The malware does not attempt to exploit the system in a traditional sense; instead, it assumes that user execution already provides sufficient access and focuses entirely on maintaining stealth while extracting maximum value.

The initial executable typically functions as a dropper. In many observed cases, this dropper is compiled in Rust, a choice that is becoming increasingly common among threat actors due to its portability, memory safety characteristics, and lower detection rates compared to more traditional languages. Upon execution, the binary performs a series of environmental checks designed to identify virtual machines, sandbox environments, or analysis tools. These checks are not overly complex, but they are sufficient to evade basic automated analysis pipelines.

After these preliminary checks, the dropper proceeds to unpack its payload. This often happens in memory or within temporary directories that are less likely to attract immediate attention. The process may involve spawning legitimate Windows processes and injecting malicious code into them, effectively blending malicious activity with normal system behavior. From a defender’s perspective, this creates a situation where malicious activity is not easily distinguishable from legitimate process execution.

At this stage, the infection chain becomes more clearly structured. The dropper prepares the system for the deployment of its primary payload, Vidar Stealer, while also setting up the secondary component, GhostSocks. The transition between these stages is intentionally quiet. There are no obvious system disruptions, no visible warnings, and no immediate signs of compromise from an end-user perspective.

6. Vidar Stealer: Data Theft at Scale

The core of this campaign revolves around Vidar Stealer, a well-established information-stealing malware that has been widely used in financially motivated operations for several years. Its continued popularity is not accidental. Vidar is designed to be fast, reliable, and highly configurable, making it an ideal tool for campaigns that rely on rapid monetization.

Once deployed, Vidar begins by profiling the system. It gathers basic information about the operating system, installed applications, and hardware configuration. While this may seem routine, it serves an important purpose. This information allows attackers to assess the value of the infected system and determine how it can best be exploited.

The primary focus, however, is on browser data. Modern browsers store a significant amount of sensitive information, including saved credentials, session cookies, and autofill data. Vidar systematically extracts this information from popular browsers such as Chrome, Edge, and Firefox. The extraction process is efficient and often completed within seconds.

What makes this particularly dangerous is the theft of session cookies. Unlike passwords, which can be protected by multi-factor authentication, session cookies represent an already authenticated state. By capturing these cookies, attackers can effectively bypass MFA and gain direct access to active sessions. This allows them to impersonate users without triggering traditional authentication challenges.

In addition to browser data, Vidar also targets cryptocurrency wallets. It searches for wallet files associated with commonly used applications and extracts them for exfiltration. Depending on the configuration, the malware may also collect files from specific directories, focusing on formats that are likely to contain sensitive information such as configuration files, documents, or exported data.

All collected data is then prepared for exfiltration. Communication with command-and-control infrastructure is typically performed over standard protocols, allowing the traffic to blend in with normal network activity. From a network monitoring perspective, this can make detection significantly more challenging, especially in environments with high volumes of outbound traffic.

7. GhostSocks: Monetizing Access Beyond Data Theft

While Vidar handles data exfiltration, GhostSocks introduces an additional layer of value for the attacker. This component transforms the infected system into a proxy node, effectively turning the victim's device into part of a distributed network that can be used for a variety of malicious purposes.

Once active, GhostSocks establishes a connection that allows external traffic to be routed through the compromised system. This traffic appears to originate from the victim's IP address, which is often associated with a legitimate residential or corporate network. This provides attackers with a significant advantage, as it allows them to bypass many forms of IP-based detection and filtering.

The implications of this are substantial. The compromised system can be used to conduct further attacks, including credential stuffing, fraud operations, and reconnaissance activities. Because the traffic is routed through legitimate networks, it is less likely to be flagged as suspicious. In effect, the victim becomes an unwitting participant in a broader criminal infrastructure.

From a monetization perspective, this capability is highly valuable. Access to residential or enterprise proxies can be sold on underground markets, providing a steady revenue stream for the attackers. This means that even if the initial data theft yields limited results, the infection can still generate ongoing value.

The combination of Vidar and GhostSocks reflects a deliberate strategy. Rather than relying on a single method of monetization, the campaign maximizes return on investment by extracting both immediate and long-term value from each compromised system.

8. Persistence and Post-Execution Behavior

Although the campaign does not rely heavily on complex persistence mechanisms, it still incorporates techniques designed to maintain access for as long as possible. These mechanisms are often subtle and may vary between samples, but they generally aim to ensure that the malware can continue operating even after a system reboot.

Persistence may be achieved through standard methods such as registry modifications or scheduled tasks. In some cases, the malware may also leverage legitimate system processes to re-establish its presence. These techniques are not particularly novel, but they are effective when combined with the overall stealth of the campaign.

What is more notable is the emphasis on rapid execution rather than long-term persistence. Vidar is designed to extract data quickly, often completing its primary objective within minutes. This reduces the need for prolonged access and minimizes the window in which the malware can be detected.

However, the presence of GhostSocks changes this dynamic. Because the proxy component benefits from sustained access, there is an incentive to maintain persistence where possible. This creates a hybrid model in which some infections are short-lived, while others persist for extended periods depending on the value of the compromised system.

9. Mapping to MITRE ATT&CK

The behavior observed in this campaign aligns closely with known adversary techniques as documented in MITRE ATT&CK. While the individual techniques are not new, their combination within a trust-based delivery model represents a notable evolution.

Initial access is achieved through user execution rather than exploitation, reflecting a shift toward social engineering within trusted environments. Execution itself is straightforward, but it is followed by techniques designed to obscure malicious activity, including process injection and the use of legitimate system binaries.

Credential access is a central component of the campaign, with Vidar extracting data directly from browser storage. This is complemented by collection and exfiltration techniques that focus on efficiency and stealth. Command-and-control communication is designed to blend in with normal traffic, reducing the likelihood of detection.

The inclusion of proxy functionality introduces an additional layer of command-and-control capability, allowing attackers to maintain indirect control over compromised systems while leveraging them for further operations.

10. Detection and Threat Hunting Considerations

Detecting this campaign requires a shift in perspective. Traditional indicators such as exploit signatures or known malicious domains may not be sufficient, particularly in the early stages of infection. Instead, defenders must focus on behavioral patterns that deviate from normal user activity.

One of the earliest and most reliable indicators is the execution of newly downloaded binaries from user directories. In developer environments, this behavior may not be uncommon, which complicates detection. However, correlating this activity with subsequent process behavior can provide valuable insights.

For example, the rapid access of browser data by a non-browser process is a strong signal of malicious intent. Similarly, the creation of network connections shortly after execution, particularly to unfamiliar or newly registered domains, can indicate data exfiltration.

The presence of proxy-like network behavior is another important indicator. Systems that suddenly begin routing unusual traffic or communicating in patterns consistent with SOCKS proxies should be investigated promptly.

From a threat hunting perspective, it is essential to look beyond individual indicators and consider the broader context. The combination of user-driven execution, rapid data access, and outbound communication forms a pattern that is difficult to attribute to legitimate activity.

11. Defensive Strategy and Security Posture Adaptation

This campaign highlights a structural gap in many security strategies. Organizations have spent years optimizing defenses against exploits, phishing campaigns, and known malware families, yet far fewer controls are designed to address threats that operate entirely within trusted workflows. The result is a blind spot where malicious activity can unfold without triggering traditional detection mechanisms.

To address this, defensive strategy must evolve beyond static controls and signature-based detection. What is required is a shift toward **behavioral awareness and contextual security**, particularly in environments where users regularly interact with external code and tools.

Developer endpoints should be treated as high-risk assets. Unlike standard user workstations, these systems frequently execute newly downloaded binaries, interact with external repositories, and maintain access to sensitive infrastructure. This combination of exposure and privilege creates an attractive target for attackers and demands a higher level of monitoring and control.

Application control plays a central role in this context. Restricting the execution of untrusted binaries, particularly from user-controlled directories such as Downloads or temporary paths, can significantly reduce the attack surface. While such controls must be implemented carefully to avoid disrupting legitimate workflows, they provide a critical safeguard against precisely the type of attack observed in this campaign.

Equally important is visibility. Organizations must ensure that they can observe not only whether a file was executed, but also how it behaves afterward. This includes tracking process relationships, monitoring access to sensitive data stores such as browser credential databases, and analyzing outbound network activity. Without this level of visibility, early-stage compromises may go unnoticed until after data has already been exfiltrated.

Another key consideration is the role of trust in external platforms. GitHub, while essential to modern development practices, should not be treated as an implicitly trusted source. Security policies should reflect the reality that malicious content can exist on legitimate platforms, and controls should be designed accordingly.

12. Detection Strategy in Practice

Detecting this campaign requires a nuanced approach that focuses on behavior rather than static indicators. The initial stages of the attack are indistinguishable from normal activity when viewed in isolation. It is only when multiple signals are correlated that a clear picture of malicious intent emerges.

One of the most effective starting points is to monitor the execution of binaries originating from user-controlled directories. While this behavior is not inherently malicious, it is often the first step in user-driven compromise scenarios. When combined with subsequent activity—such as unexpected access to browser data or the creation of outbound connections—it becomes a much stronger indicator.

The interaction between processes is another valuable source of insight. Legitimate applications tend to follow predictable patterns of behavior, whereas malware often introduces anomalies. For example, a newly executed binary that rapidly spawns additional processes or injects into existing ones should be considered suspicious, particularly if this activity is followed by network communication.

Network behavior provides further opportunities for detection. Although the malware attempts to blend in with normal traffic, certain patterns can still be identified. Systems that begin communicating with unfamiliar domains shortly after executing a new binary, or that exhibit characteristics consistent with proxy activity, warrant closer investigation.

It is also important to consider the broader context in which these signals occur. A single indicator may not be sufficient to justify action, but a combination of indicators—especially when aligned with known campaign characteristics—can provide a high-confidence detection. This underscores the importance of integrated security platforms that can correlate data across endpoints, networks, and identity systems.

13. Incident Response Considerations

When a potential compromise related to this campaign is identified, time becomes a critical factor. The design of Vidar Stealer prioritizes speed, meaning that sensitive data may already have been exfiltrated within minutes of execution. As a result, incident response efforts must focus not only on containment but also on assessing the extent of exposure.

The first priority is to isolate the affected system. This prevents further data exfiltration and disrupts any ongoing communication with command-and-control infrastructure. At the same time, it is important to preserve relevant evidence, including memory and disk artifacts, to support investigation and potential attribution.

Credential security must be addressed immediately. Because the malware targets browser-stored credentials and session data, it is not sufficient to reset passwords alone. Active sessions must be invalidated, and tokens revoked, to ensure that attackers no longer have access. This is particularly important for cloud services, where session hijacking can provide persistent access even after credentials are changed.

The scope of the investigation should extend beyond the initial endpoint. Stolen credentials may have already been used to access additional systems, meaning that lateral movement is a real possibility. Reviewing authentication logs, access patterns, and recent activity across critical systems can help identify signs of unauthorized access.

In cases where GhostSocks is present, additional attention should be paid to network activity. The compromised system may be participating in proxy operations, which can have implications for both security and compliance. Identifying and terminating these connections is an essential part of the response process.

Recovery efforts should be approached with caution. Given the potential for widespread credential exposure, organizations may need to implement broader remediation measures, including forced credential resets and enhanced monitoring. The goal is not only to restore normal operations but also to ensure that any residual access has been fully eliminated.

14. Strategic Outlook

The trojanized “Claude Code” campaign is not an isolated incident. Rather, it represents a clear example of how threat actors are adapting to changes in technology and user behavior. As AI tools and developer ecosystems continue to grow, they will increasingly become targets for exploitation.

One of the most significant trends highlighted by this campaign is the shift toward **event-driven attacks**. Instead of relying on long-term planning or complex exploitation, attackers are capitalizing on real-time developments. This allows them to align their campaigns with moments of heightened interest and reduced skepticism, increasing their chances of success.

Another important trend is the continued rise of **trust-based attack vectors**. By operating within platforms that users already trust, attackers can bypass many of the controls designed to detect and prevent malicious activity. This challenges traditional security models and requires a reevaluation of how trust is defined and enforced.

The use of commodity malware such as Vidar further reinforces the idea that sophistication is not always necessary for effectiveness. Well-established tools, when combined with strong social engineering and timely execution, can achieve results comparable to more advanced threats. This lowers the barrier to entry for attackers and increases the frequency of such campaigns.

Looking ahead, it is likely that similar techniques will be applied to other areas of the technology landscape. Any platform that combines high user trust with the ability to distribute executable content is a potential target. This includes not only code repositories but also package managers, collaboration tools, and developer marketplaces.

15. Conclusion

The trojanized “Claude Code” campaign demonstrates how effectively threat actors can exploit the intersection of trust, timing, and user behavior. By leveraging a real-world event and embedding themselves within a trusted platform, attackers were able to deliver malware without relying on traditional intrusion techniques.

The use of Vidar Stealer and GhostSocks reflects a pragmatic approach focused on maximizing value from each compromised system. Credential theft, session hijacking, and proxy-based monetization combine to create a flexible and resilient attack model that can adapt to different environments and objectives.

For organizations, the key takeaway is clear. Security strategies must evolve to address threats that operate within legitimate workflows and trusted platforms. This requires greater emphasis on behavioral detection, improved visibility into user activity, and a more critical approach to external sources of software.

Ultimately, the success of this campaign is not a reflection of technical weakness, but of misplaced trust. Addressing this challenge will require not only new tools and controls, but also a shift in mindset—one that recognizes that even the most familiar environments can be used as entry points for compromise.

16. Sources and References

The findings and analysis presented in this report are based on a combination of open-source intelligence (OSINT), industry research, and observed campaign patterns consistent with known malware distribution techniques. Particular emphasis has been placed on corroborating technical details across multiple reputable sources to ensure accuracy and relevance.

A key reference for this campaign is reporting on the rapid weaponization of the Claude Code exposure, as documented by The Register. Their coverage highlights how threat actors leveraged trojanized GitHub repositories to distribute malware under the guise of leaked or enhanced versions of the tool. The article provides valuable context regarding the speed of exploitation and the use of Vidar Stealer in conjunction with proxy-based malware.

Further technical insight into the campaign and its broader implications within the AI ecosystem can be found in research published by Trend Micro. Their analysis explores how attackers are increasingly targeting developer communities and weaponizing trust in widely used platforms. This aligns closely with the behavioral patterns observed in this campaign.

Additional context on the initial Claude Code exposure and its security implications has been discussed by Zscaler, which examined how the leak created an opportunity for malicious actors to exploit heightened interest and curiosity among developers.

The behavior and capabilities of Vidar Stealer referenced in this report are consistent with prior analyses from multiple threat intelligence providers. Background information on Vidar, including its data exfiltration techniques and role within Malware-as-a-Service ecosystems, is well documented by organizations such as Microsoft and Kaspersky. These sources provide detailed breakdowns of credential theft mechanisms, browser data extraction, and command-and-control communication patterns.

The mapping of observed techniques to adversary behavior frameworks is based on the MITRE ATT&CK framework. This widely adopted model enables consistent classification of tactics such as user execution, credential access, and data exfiltration, and supports alignment between detection strategies and known threat behaviors.

In addition to these primary sources, this report incorporates general threat intelligence principles and observed patterns from recent campaigns targeting developer ecosystems, including the abuse of platforms such as GitHub for malware distribution. These patterns have been repeatedly documented across the industry and are increasingly recognized as a significant attack vector.

Referenced Materials

The following publicly available resources were used to support this report and validate the observed campaign behavior, malware characteristics, and threat landscape context:

- The Register – *Trojanized "Claude Code" GitHub repositories spreading malware* https://www.theregister.com/2026/04/02/trojanized_claude_code_leak_github/
- Trend Micro – *Weaponizing Trust: Claude Code Lures and GitHub Payload Distribution* https://www.trendmicro.com/en_us/research/26/d/weaponizing-trust-claude-code-lures-and-github-release-payloads.html

- Zscaler ThreatLabz – *Anthropic Claude Code Exposure and Security Implications*
  https://www.zscaler.com/blogs/security-research/anthropic-claude-code-leak
- MITRE – *ATT&CK Framework (Enterprise Matrix)*
  https://attack.mitre.org/
- Microsoft Security Intelligence – *Information Stealers and Credential Theft Techniques*
  https://learn.microsoft.com/en-us/microsoft-365/security/intelligence/
- Kaspersky – *Vidar Stealer Malware Analysis and Capabilities*
  https://securelist.com/

Analytical Note

While specific infrastructure indicators (such as domains or file hashes) may evolve rapidly in campaigns of this nature, the underlying techniques and operational patterns remain consistent. The references above therefore serve not only as documentation of this specific campaign, but also as a broader foundation for understanding similar threats.

17. Our Perspective: How Ransomwared Helps

Modern infostealer operations expose a fundamental weakness in many contemporary security strategies: an overreliance on the assumption that malicious activity will be noisy, persistent, or technically complex. Infostealers deliberately violate that assumption. They are brief, quiet, and designed to disappear before defenders have time to react. Their true impact unfolds later, when stolen credentials, sessions, and tokens are used to access systems in ways that appear legitimate. This is the gap where many organizations lose visibility — and it is precisely the gap Ransomwared was built to address.

At **Ransomwared**, we start from a different premise than most security tools. We assume that initial access will happen. We assume that endpoint defenses may not always see it. And we assume that attackers will increasingly rely on legitimacy rather than malware to achieve their objectives. Instead of asking how to block every possible infostealer variant, we ask a more practical and more urgent question: *How do we detect and respond when stolen access is being prepared for abuse?*

Seeing What Traditional Tools Miss

Traditional endpoint security solutions are optimized to detect malicious code execution, persistence mechanisms, and exploit behavior. These controls remain essential, but infostealers are explicitly designed to operate beneath their threshold of concern. A short-lived user-context process that reads browser files and exits may never rise above a low-severity alert, if it is flagged at all.

Ransomwared focuses on the *meaning* of that activity rather than its superficial characteristics. A process accessing browser credential stores is not dangerous because it is malware; it is dangerous because it is extracting the very artifacts that grant legitimate access to the enterprise. By treating credential harvesting as a first-class security event rather than a peripheral signal, Ransomwared helps organizations recognize the moment when an environment becomes vulnerable to identity abuse and ransomware staging.

Bridging Endpoint, Identity, and Behavior

One of the defining challenges of infostealer-driven intrusions is fragmentation of visibility. Endpoint tools see isolated file access. Identity platforms see successful logins. Cloud services see authorized actions. Each system, viewed independently, may show nothing overtly malicious.

Ransomwared is designed to bridge these domains. It correlates endpoint behaviors with identity activity and higher-level behavioral patterns, allowing defenders to see the full narrative of an intrusion rather than disconnected fragments. This correlation is critical in identifying the transition from initial access to pre-impact staging — the phase where attackers prepare for ransomware deployment, data exfiltration, or long-term abuse.

By focusing on patterns rather than signatures, Ransomwared remains effective even as infostealer families change. The specific malware name is less important than the sequence of actions that follows: credential harvesting, session reuse, reconnaissance, and preparation. These patterns are remarkably consistent across campaigns, even as tooling evolves.

Detecting the Ransomware On-Ramp

Infostealers are rarely the end goal. They are the on-ramp to more destructive activity. Ransomwared is built to identify that on-ramp early, before encryption or extortion occurs.

This includes monitoring for behaviors that commonly precede ransomware deployment, such as unusual enumeration of systems, access to backup infrastructure, and changes to security-relevant configurations. Even

when attackers operate entirely through legitimate interfaces, these preparatory actions create subtle but detectable deviations from normal behavior.

By surfacing these signals early, Ransomwared gives organizations time — time to revoke access, contain affected accounts, and disrupt the attack before it reaches the impact stage. In a landscape where minutes or hours can make the difference between a contained incident and a full-scale outage, this early warning capability is decisive.

Designed for Resilience, Not Replacement

Ransomwared is not positioned as a replacement for existing security investments. EDR, identity protection, and SIEM platforms remain foundational components of modern defense. However, infostealer-driven attacks demonstrate that no single layer can be relied upon exclusively.

Instead, Ransomwared acts as a resilience layer. It assumes that other controls may be bypassed or temporarily blinded and focuses on maintaining visibility when attackers believe they are operating undetected. This philosophy mirrors the reality of modern attacks, where success often depends on exploiting gaps between tools rather than defeating them outright.

In practice, this means Ransomwared complements existing telemetry rather than duplicating it. It adds context, correlation, and behavioral insight that help defenders make sense of ambiguous signals and act decisively under uncertainty.

Supporting Defenders, Not Overwhelming Them

Another critical design principle is usability for security teams. Infostealer-driven intrusions generate ambiguity rather than obvious alerts. Analysts are often faced with incomplete information and difficult decisions about whether to escalate or wait.

Ransomwared is designed to reduce this uncertainty. By highlighting high-risk behavioral patterns and linking them to likely attacker objectives, it helps analysts prioritize what matters most. The goal is not to generate more alerts, but to generate *clearer* ones — alerts that explain why an event is concerning and what kind of impact it may enable if left unaddressed.

This clarity is especially important in environments with limited SOC resources. When every investigation competes for attention, understanding which signals represent genuine pre-impact activity is essential.

Aligning With the Reality of Modern Attacks

The broader lesson of modern infostealer operations is that attackers are no longer trying to outsmart defenses at the technical level alone. They are exploiting trust, convenience, and architectural assumptions. They are operating where defenders are least likely to look: in successful logins, authorized actions, and legitimate workflows.

Ransomwared is built for this reality. It is grounded in the assumption that legitimacy can be abused and that visibility must extend beyond traditional malware indicators. By focusing on behavior, access, and intent rather than code alone, it provides organizations with a fighting chance against threats that are designed to blend in.

Keeping the Lights On When Others Go Dark

Ultimately, the value of Ransomwared lies in its ability to maintain visibility when attackers believe they have achieved invisibility. Infostealers create the illusion of normalcy — a quiet environment where everything appears to be functioning as intended until it suddenly is not. Ransomwared challenges that illusion by watching for the subtle signs that normal activity has crossed into malicious preparation.

In a threat landscape where infostealers have become the connective tissue between initial compromise and enterprise-scale impact, this capability is no longer optional. It is a requirement for organizations that want to detect attacks before they reach the point of no return.

The reality is clear: modern attacks do not always announce themselves. They often arrive quietly, dressed in legitimacy, and wait patiently for the right moment to strike. Ransomwared exists to ensure that even in those moments, defenders are not operating in the dark.