HeadCrab 2.0: Evolving Threat in Redis Malware Landscape

By Asaf Eitani

Published: 2024-01-29 · Archived: 2026-04-05 13:26:34 UTC

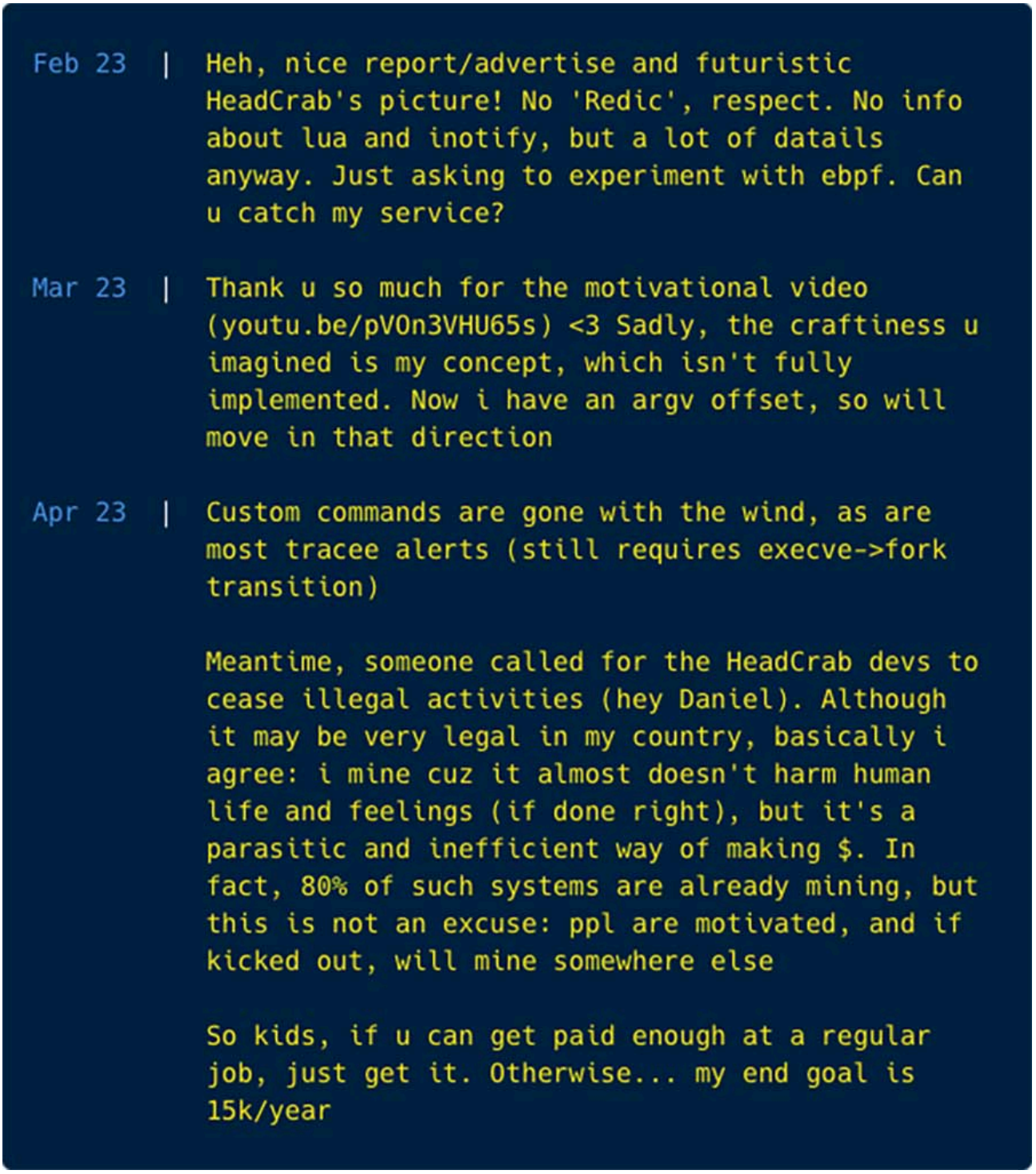
At the beginning of 2023, Aqua Nautilus researchers uncovered HeadCrab – an advanced threat actor utilizing a state-of-the-art, custom-made malware that compromised 1,200 Redis servers. As you know in the ever-evolving world of cybersecurity, threat actors continually adapt and refine their techniques. Recently, our researchers detected a new version of the HeadCrab malware targeting our honeypots. This blog post delves into the intricate details of HeadCrab 2.0, revealing its advanced mechanisms, our ongoing efforts to combat this sophisticated threat, and shows that one year later the campaign has almost doubled the number of infected Redis servers.

The Attacker's Mini Blog: A Closer Look

Inside the malware, the attacker behind HeadCrab is managing a 'mini blog', a small text discussing the developments of the malware, strategies of the campaign and specific references to related events. This blog has become a solid source of information, providing us with insights directly from the attacker's perspective. In the first version of HeadCrab we noticed a mention of Aqua in this mini blog, referencing a prior blog post we published. In this new version of the malware, the threat actor mentioned us again, referencing our discovery and analysis of his mischief, claiming that he can bypass our eBPF solution.

Spoiler alert – he fails to do so.

Below you can see his reference to our research, alongside other interesting pieces of information:



Key Takeaways from the Mini Blog

1. **Acknowledgment of Our Research**: The attacker has explicitly referenced our previous blog post on the first version of HeadCrab. This acknowledgment shows that our research has gained significant attention, influencing even those whom we seek to thwart.
2. **Reference to External Coverage**: The mini blog also references a YouTube video by security researcher Daniel Lowrie, which covered our findings on HeadCrab. This indicates the attacker's awareness of the broader cybersecurity community's response to their activities.

3. **Adaptation Strategies**: The blog entries highlight the attacker's efforts to evolve the malware, specifically to evade our open source detection tool, Tracee. This is a clear indication of the malware's adaptability and the attacker's commitment to staying ahead of security measures.
4. **Enhanced Defense Evasion Techniques**: The mini blog details specific changes in the malware's operation, particularly in how it communicates, and controls compromised systems. These changes were done to enhance the defense evasion capabilities of the malware and ensure the campaign remains hidden.

The Implications of the Attacker Engagement

This engagement is more than just a communication from the attacker; it's a strategic move with several implications:

1. **Heightened Awareness**: The attacker's acknowledgment of our work illustrates the learning curve of advanced attackers who optimize their tactics, techniques and procedures like a chess game, according to the move of their adversaries (the security industry).
2. **Improved Defense Evasion**: Another aspect that is depicted in this blog is how the threat actor is trying to improve the defense evasion techniques in order to better conceal the campaign.
3. **Valuable Intelligence**: The information shared by the attacker, while potentially misleading, is a goldmine for threat intelligence. It offers a rare peek into the mindset and tactics of the adversary, which can be leveraged to enhance our defensive measures.
4. **Evolving Threat Landscape**: The willingness of attackers to directly engage with security researchers signifies a dynamic and evolving threat landscape. It reflects the ongoing cat-and-mouse game between attackers and defenders in cybersecurity.

This engagement from the HeadCrab attacker presents both a challenge and an opportunity. It underscores the importance of our work at Aqua Security and reinforces the need for continuous vigilance and innovation in our defense strategies. As we dissect the technical intricacies of HeadCrab 2.0, this engagement will serve as a backdrop, reminding us of the ever-present and evolving nature of cyber threats.

Technical Analysis of HeadCrab 2.0: Unraveling the Advanced Malware

In our continued efforts to understand and mitigate cyber threats, we've conducted an in-depth technical analysis of HeadCrab 2.0. This new version exhibits advancements over its predecessor, showcasing the attacker's increasing sophistication in malware development.

Enhanced Evasion Techniques in HeadCrab 2.0: Fileless Loader Mechanism

An integral aspect of the sophistication of HeadCrab 2.0 lies in its advanced evasion techniques. In contrast to its predecessor (named HeadCrab 1.0), this new version employs a fileless loader mechanism, demonstrating the attacker's commitment to stealth and persistence.

HeadCrab 1.0 – No Fileless loader

In the previous version, the attacker utilized the `SLAVEOF` command to download and save the HeadCrab malware .so (shared object) file to disk. This method, while effective, left tangible traces on the file system, making it

susceptible to disk scanning solutions and easier for cybersecurity defenses to detect and mitigate.

HeadCrab 2.0 – The Fileless Loader

The new attack vector involves the use of a loader .so file. This loader, instead of directly saving the HeadCrab malware on the disk, receives the malware's content over the Redis communication channel and stores it in a fileless location. By opting for a fileless storage approach, HeadCrab 2.0 significantly reduces its digital footprint on the affected host. This method effectively circumvents traditional disk-based scanning solutions, making the malware much harder to detect. The fileless technique ensures that the malware leaves minimal traces on the infected system. This subtlety not only aids in evasion but also complicates forensic analysis and threat hunting efforts, as the usual file-based indicators of compromise are absent.

Command and Control (C2) Channel Evolution

A critical aspect of HeadCrab 2.0 is its evolved Command and Control (C2) communication strategy, marking a departure from the earlier versions approach.

HeadCrab 1.0's Strategy – custom commands

The original HeadCrab malware used custom Redis commands (rds*) for its C2 interactions. This method, while effective, made the malware somewhat easier to detect due to the presence of these unusual commands.

HeadCrab 2.0's Strategy – default commands

The new version cunningly uses the default MGET command in Redis. By hooking into this standard command, the malware gains the ability to control it during specific attacker-initiated requests. Those requests are achieved by sending a special string as an argument to the MGET command. When this specific string is detected, the malware recognizes the command as originating from the attacker, triggering the malicious C2 communication. For regular users, the MGET command functions as expected, thereby maintaining the stealth of the malware.

Detection Challenges and Strategies

With the evolution of HeadCrab 2.0, our previous detection methods required a significant overhaul.

HeadCrab 1.0

Initially, compromised servers were identified by executing the COMMAND command and looking for custom rds* commands. This method was rendered ineffective with the new version's stealthier approach.

HeadCrab 2.0

We discovered a flaw in the hooked function of the CONFIG command in HeadCrab 2.0. The malware responds with an +OK to commands like CONFIG SETREWRITE DIRDBFILENAME, which could result in a different response in uncompromised systems.

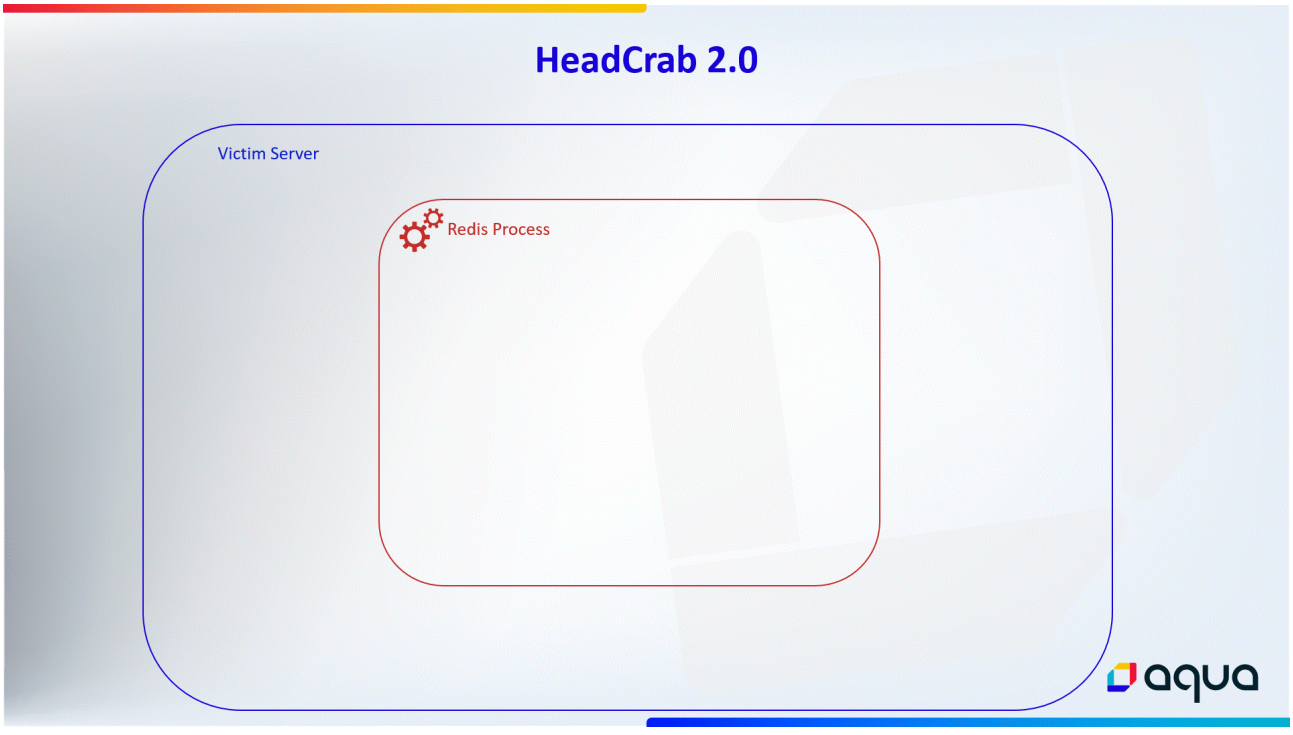
Global Scan for Compromised Servers

We conducted a global scan, executing commands designed to change the *DIR* key to a non-existent path. Normal servers responded with an error, while compromised servers, affected by HeadCrab 2.0, returned an +OK response. This method proved effective in identifying an additional 1,100 compromised servers.

Highlighting HeadCrab's Advanced Hooking Method

In HeadCrab 2.0 we observed the redisCommandProc pointer within the redisCommand structure. This pointer is redirected to a function that the attacker controls. This method is subtle and harder to detect as it involves lower-level manipulation of data structures within Redis and harness the Redis framework inner workings to the benefit of the threat actor.

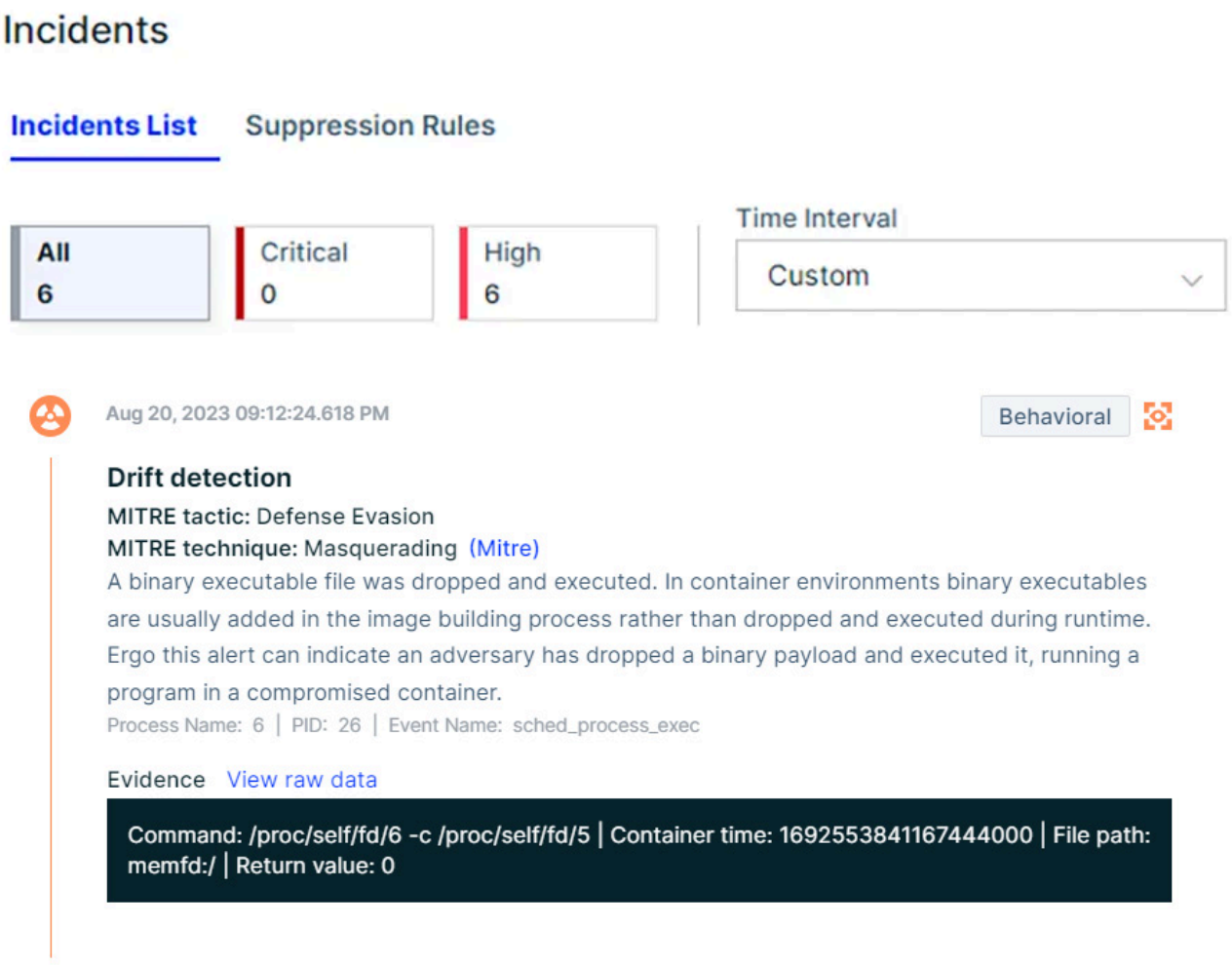
In summary, HeadCrab 2.0 represents an escalation in the sophistication of Redis malware. Its ability to hide in plain sight, masquerading its malicious activities under standard commands, poses new challenges for cybersecurity experts. At Aqua Security, we continue to adapt our methods and tools to detect and counter such advanced threats, ensuring the security and integrity of systems globally.



Conclusion: Navigating the Evolving Threat of HeadCrab 2.0

The emergence of HeadCrab 2.0 represents a significant milestone in the ever-changing landscape of cybersecurity threats. Our analysis not only highlights the technical sophistication of this new version but also emphasizes the dynamic nature of cyber threats and the need for continuous adaptation in security strategies. At Aqua Security, we are dedicated to staying ahead of these threats, developing cutting-edge solutions to protect against such sophisticated and elusive malware.Key Insights and Implications

1. **Adaptive and Stealthy Malware**: HeadCrab 2.0 showcases an advanced level of adaptability and stealth. Its ability to masquerade malicious activities under standard Redis commands poses a challenging scenario for cybersecurity professionals. This evolution underscores the necessity for continuous research and development in security tools and practices.
2. **Importance of Vigilant Monitoring**: The engagement by the attacker and the subsequent evolution of the malware highlights the critical need for vigilant monitoring and intelligence gathering. Staying abreast of threat actors' tactics and techniques is crucial for timely and effective response.
3. **Collaborative Defense**: The acknowledgment of our work by the attacker and the broader community's interest in our findings underscore the importance of collaborative efforts in the cybersecurity field. Sharing knowledge and strategies across organizations and experts is key to building a robust defense against such sophisticated threats.
4. **Evolving Detection Methods**: Our journey in detecting and analyzing HeadCrab 2.0 demonstrates the need for evolving detection methods. Aqua's CNDR is in the cutting edge of behavioral detection, fed by advanced threats like HeadCrab researched by us, Aqua Nautilus.



Aug 20, 2023 09:12:24.618 PM

Behavioral

Fileless execution detected

MITRE tactic: Defense Evasion
MITRE technique: Reflective Code Loading
Fileless execution was detected. Executing a process from memory instead from a file in the filesystem may indicate that an adversary is trying to avoid execution detection.
Process Name: 6 | PID: 26 | Event Name: sched_process_exec

Evidence View raw data

```
Command: /proc/self/fd/6 -c /proc/self/fd/5 | File path: memfd:/ | Return value: 0
```

Aug 20, 2023 09:12:35.801 PM

Behavioral

New executable was dropped to memory during runtime

MITRE tactic: Defense Evasion
MITRE technique: Reflective Code Loading
Fileless Loading ELF file written straight into memory. Executing a process from memory instead from a file in the filesystem may indicate that an adversary is trying to avoid execution detection.
Process Name: redis-server | PID: 39 | Event Name: magic_write

Evidence View raw data

```
File path: /dev/shm/pke
```

Aug 20, 2023 09:12:38.509 PM

Behavioral

New executable was dropped to memory during runtime

MITRE tactic: Defense Evasion
MITRE technique: Reflective Code Loading
Fileless Loading ELF file written straight into memory. Executing a process from memory instead from a file in the filesystem may indicate that an adversary is trying to avoid execution detection.
Process Name: redis-server | PID: 41 | Event Name: magic_write

Evidence View raw data

```
File path: /dev/shm/h
```

Asaf is a Security Researcher at Aqua Nautilus research team. He focuses on researching Linux malware, developing forensics tools, and analyzing new attack vectors in cloud native environments. In his spare time, he likes painting, playing beach volleyball, and carving wood sculptures.

Assaf Morag

Assaf is the Director of Threat Intelligence at Aqua Nautilus. He is responsible of acquiring threat intelligence related to software development life cycle in cloud native environments, supports the team's data needs, and helps Aqua and the ecosystem remain at the forefront of emerging threats and protective methodologies. His research has been featured in leading information security publications and journals worldwide, and he has presented at leading cybersecurity conferences. Notably, Assaf has also contributed to the development of the new MITRE ATT&CK Container Framework.
Assaf is leading an O'Reilly course, focusing on cyber threat intelligence in cloud-native environments. The course covers both theoretical concepts and practical applications, providing valuable insights into the unique challenges and strategies associated with securing cloud-native infrastructures.
