RAM Scraping - what it is, how attackers lift data from memory, and how to prevent it

Gridinsoft Help Center

What it is

RAM scraping is when malware reads a process's live memory to grab sensitive data in plaintext before it's encrypted or after it's decrypted. Classic targets are point-of-sale apps where payment card data briefly appears in RAM, but attackers also scrape browsers, password managers, and system processes to lift logins, session tokens, API keys, and other secrets.

Why it matters

Data in memory can bypass at-rest and in-transit protections. One infected checkout lane or server can leak thousands of cards or hand over high-privilege sessions for lateral movement.

How it works

- Malware injects into or attaches to a target process.
- Scans memory for patterns like PANs, Track 1/2, CVV, cookies, tokens, or form fields.
- Copies matches into buffers, optionally compresses or encrypts them.
- Exfiltrates to attacker infrastructure on a schedule.

Red flags

- Processes repeatedly calling ReadProcessMemory or creating suspicious memory dumps.
- Unusual outbound connections from POS or kiosk networks.
- EDR alerts on code injection, credential access, or memory scraping behavior.
- Sudden spikes in declined cards or fraud tied to a specific location.

Prevent it

- Use point-to-point encryption and tokenization on POS so card data never appears in cleartext.
- Lock down endpoints: application allowlisting, least privilege, disable macros, patch aggressively.
- Protect credentials: browser hardening, disable unnecessary password storage, monitor for cookie theft.
- Segment payment networks and enforce strict egress filtering and TLS inspection where allowed.
- Monitor for memory-access APIs misuse, code injection, and unexpected dumps; respond and

reimage if compromised.