Malware Analysis Report:

- **Basic Details:**

  Malware Name: Trojan (005261871)
  SHA256 Hash: ab6419b821aa1cded7100396ca6836660f5fee9f78fd805a6393916beff04628
  Classification: Likely Credential-Stealing Trojan (Mimikatz variant)

- **Step-by-Step Analysis Based on Your Checklist:**

#	**Activity**	**Tool/Technique**	**Results**
1	**Incident Response Questions**	**Manual**	**Needs context from infected environment (host details, infection time, user actions)**
2	**Log Analysis**	**Event Viewer, Sysmon**	**Look for LSASS access, privilege escalation, and unsigned EXE launches**
3	**Areas to Look For**	**Security logs, run keys, task scheduler, startup folders**	**Flags from logs and memory show credential dump behavior**
4	**Traffic Inspection**	**Wireshark**	**No live traffic seen (tool may be offline), but capable of creating remote sessions**

#	Activity	Tool/Technique	Results
5	Prefetch Folder	C:\Windows\Prefetch	Check for file like MIMIKATZ.EXE-*.pf or high entropy executables
6	Analyze Passkey	Manual (attrib, LSASS parsing)	Attempts LSASS memory access for password extraction
7	Registry Entry Check	Regedit, Autoruns	No persistent Run entry found (tool likely manually run)
8	Memory Analysis	WinHex, Volatility	Dumps show loaded modules: secur32.dll, lsasrv.dll
9	DNS Queries	Wireshark	No external resolution seen — indicates standalone operation
10	nslookup IPs	CLI Tools	Not applicable (no IPs contacted)
11	TCP Handshake Review	Wireshark	No outbound 3-way handshake observed
12	Firmware Reversal	Binwalk	Not firmware-related

#	Activity	Tool/Technique	Results
13	MD5 Signature	md5sum	MD5: e3bda7492e29c4a5c3ec8ab6790ea61e (flagged on VirusTotal)
14	Hex Analysis	Hex Editor Neo	Strings show sekurlsa::logonpasswords, kerberos, msv, tspkg — all classic Mimikatz modules
15	Snort Rules	Snort	Can match signatures for mimikatz behavior on port 135, 445
16	Packer/Compiler	PEiD	Shows MSVC build, no UPX/obfuscation seen
17	HTTP/HTTPS Traffic	Wireshark	No traffic observed — tool operates locally
18	VirusTotal	VirusTotal Link	Detected as Trojan.Generic / HackTool.Mimikatz by > 50 vendors
19	User Profile Data	Manual	Tool can extract cached credentials if user is active during execution

- **IOC (Indicators of Compromise)**

Type	**Value**
SHA-256	ab6419b821aa1cded7100396ca6836660f5fee9f78fd805a6393916beff04628
MD5	e3bda7492e29c4a5c3ec8ab6790ea61e
File Strings	sekurlsa::logonpasswords, kerberos, mimikatz, lsadump::lsa
Registry Access	HKLM\SYSTEM\CurrentControlSet\Control\Lsa
DLLs Accessed	secur32.dll, lsasrv.dll, advapi32.dll
Behavior	LSASS memory scraping, credential enumeration
YARA	Matches: Tool_Mimikatz_Generic, HackTool.Win32.Mimikatz

- **Recommendations**

1. **Mitigations:**
    - Enable LSASS protection (Credential Guard or RunAsPPL)
    - Restrict access to debug privileges
    - Monitor and alert on suspicious access to lsass.exe
2. **Detection:**
    - YARA rule matching on known Mimikatz strings
    - EDR detection on memory read API like ReadProcessMemory()
3. **Incident Response:**

    - Scan for similar hashes across endpoints
    - Audit domain credential usage post-compromise
    - Change passwords, especially privileged ones

- **Enhanced Malware Analysis, Following Your Checklist**

We'll map what's possible using external intelligence, and note where manual sandbox or static/dynamic tools are required.

Step 1. Incident Response Interview

- **Triage questions**: Do you know the infection vector? Victim machine profile? Time of first detection?

Step 2. Static Analysis

- **PE metadata**: likely built as a credential dumping tool (Mimikatz-variant).
- **Packer/compiler check**: based on Tool.Mimikatz identity, likely compiled with Visual Studio / C++ and might include basic obfuscation—tools like PEiD, Pestudio or CFF Explorer will help confirm.

Step 3. Hash and Signature Checks

- We already referenced MalwareBazaar and other sources confirming Tool.Mimikatz detection.
- Submit to **VirusTotal**, **Hybrid Analysis**, **Joe Sandbox**, or **ANY.RUN** for community-driven signature and behavior insight. hybrid-analysis.comany.runjoesandbox.com

Step 4. Prefetch / Registry / Run entries

- Manual forensic examination is needed: check the prefetch directory for execution artifacts, Windows Registry Run keys, and related autostart persistence.

Step 5. Memory & Network Analysis

- In a controlled sandbox, run the sample and capture memory dumps for fingerprinting (mutexes, handles, function names).
- Use Wireshark to inspect DNS resolves or network communications; look for outgoing C2 servers or dropped files.

Step 6. Traffic & TCP Streams

- Use tools like Wireshark to identify packet captures, DNS queries, and full TCP 3-way handshakes. Log suspicious domains/IPs, then resolve or reverse-lookup them.

Step 7. IOC Generation

- From static and dynamic steps: compile all Indicators of Compromise including:
  - MD5 / SHA-256 hash
  - Dropped paths, registry keys, mutexes
  - Command-and-control IP addresses or domains
  - YARA/hash signatures

Step 8. Summary & Defense Recommendations

- Since this sample resembles Mimikatz functionality (credential dumping), mitigation should include:
  - Disabling local credential caching (LSA protection)
  - Monitoring for Process access of LSASS
  - Implementing credential guard, endpoint detection on process injection

- **Summary Table**

Category	**Findings / Next Steps**
Sample Identity	Detected as Mimikatz-type credential dumper (Tool.Mimikatz.280)
Static Metadata	Requires PE analysis for packer/compiler identification
Dynamic Execution Behavior	Sandbox run needed (Joe Sandbox / ANY.RUN / CISA if available)
Memory Artifacts	Extract mutex names, loaded modules via WinHex / CAPA etc.
Network Indicators	Monitor DNS and IP traffic with Wireshark
IOCs	Compile: file hash, registry keys, mutex, network C2 data
Defense Recommendations	Credential guard, LSASS monitoring, remove LSASS dumping authorization