INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-14 13:49 UTC

Secret Blizzard's Kazuar Botnet: A Three-Tier Architecture Built to Outlast Detection

THREAT CAMPAIGN | **HIGH** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0312
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	9.5
Affected Products	Windows environments (named pipe, mailslot, COM objects, .NET loader); Microsoft Exchange (EWS-based C2); government and diplomatic targets in Europe, Central Asia, and Ukraine
Published	2026-05-14T15:00:00+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

Microsoft Threat Intelligence has disclosed a detailed architectural analysis of Kazuar, an advanced espionage implant operated by Secret Blizzard, a Russian GRU-affiliated threat group. Kazuar has evolved into a modular peer-to-peer botnet with a three-tier node architecture designed to minimize network visibility and outlast traditional detection methods. Organizations operating government, diplomatic, or critical infrastructure environments in Europe, Central Asia, and Ukraine face the highest exposure; the implant's abuse of Microsoft Exchange as a C2 channel means compromise may already exist in environments without behavioral monitoring.

Technical Analysis

Kazuar is a Windows implant attributed to Secret Blizzard (also tracked as Turla) with a three-tier peer-to-peer botnet architecture: Kernel nodes maintain persistent footholds, Bridge nodes relay and obfuscate C2 traffic, and Worker nodes execute tasked operations. A leader election mechanism designates a single node to handle all external C2 communications, routing traffic for the rest of the botnet and dramatically reducing the observable network footprint. Intra-botnet communication uses Windows named pipes, mailslots, and COM objects. External C2 abuses Microsoft Exchange Web Services (EWS), a legitimate protocol channel that many organizations do not inspect. The implant supports 150+ configurable parameters and loads via a .NET loader. Relevant CWEs: CWE-284 (Improper Access Control), CWE-923 (Improper Restriction of Communication Channel), CWE-506 (Embedded Malicious Code). MITRE techniques include T1071.003 (Application Layer Protocol: Mail Protocols), T1572 (Protocol Tunneling), T1090.001 (Internal Proxy), T1055 (Process Injection),

T1573.001 (Encrypted Channel: Symmetric Cryptography), T1547 (Boot or Logon Autostart Execution), T1497 (Virtualization/Sandbox Evasion), and T1059.001 (PowerShell), among others. No CVE is associated with this campaign; the implant exploits architecture and legitimate protocol abuse rather than a patchable vulnerability. Source: Microsoft Security Blog (2026-05-14).

Action Checklist

1. Containment: Audit Microsoft Exchange environments for anomalous EWS activity immediately. Block or restrict EWS access to known, authorized clients using Exchange throttling policies and network controls. Isolate any Windows hosts exhibiting unexplained named pipe or mailslot activity pending investigation.
2. Detection: Query Exchange transport logs and IIS logs for atypical EWS client patterns, especially repeated programmatic access from internal hosts that are not mail clients. Use Sysmon Event IDs 17 and 18 to detect named pipe creation and connection events. Hunt for .NET loader artifacts and COM object registration changes on endpoints in government, diplomatic, or critical infrastructure segments. Correlate with MITRE T1071.003 and T1572 behavioral signatures in your SIEM.
3. Eradication: There is no security patch to apply; Kazuar exploits architecture and legitimate protocol channels, not a patchable vulnerability. Eradication requires full forensic investigation of affected hosts, removal of the Kazuar implant and its persistence mechanisms (autostart entries per T1547), and revocation of any credentials present on compromised systems. Engage Microsoft Incident Response or a qualified DFIR team for confirmed infections.
4. Recovery: Validate EWS access controls post-remediation and confirm named pipe and mailslot activity has returned to baseline. Re-image confirmed Kernel and Worker nodes rather than attempting in-place cleanup. Monitor Exchange EWS logs and endpoint telemetry for at least 30 days post-remediation given the botnet's leader election capability, which allows surviving nodes to reconstitute C2.
5. Post-Incident: Review whether behavioral detection rules for protocol tunneling (T1572) and internal proxy use (T1090.001) exist and are tuned. Assess whether EWS is required in your environment; disable it if not. Evaluate whether endpoint telemetry (Sysmon or equivalent) covers named pipe and COM object activity. This campaign exposes gaps in monitoring legitimate protocol channels for C2 abuse.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to senior DFIR and legal/privacy counsel if forensic analysis confirms Kazuar Kernel node presence on Exchange infrastructure, if credential material (NTLM hashes, Kerberos tickets) was accessible to compromised nodes, or if the environment is subject to NIS2, GDPR, or sector-specific breach notification obligations in EU or Ukraine jurisdictions given Secret Blizzard's government and diplomatic targeting profile.

Recovery Notes	Re-image all confirmed Kazuar Kernel and Worker nodes rather than attempting in-place remediation, as Kazuar's in-memory operation and modular loader make complete on-disk artifact removal unreliable. Post-remediation, monitor Exchange EWS IIS logs and Sysmon named pipe telemetry for a minimum of 30 days, as Kazuar's leader election mechanism allows surviving Tier-2 or Worker nodes to autonomously elect a new Kernel node and reconstitute C2 without external attacker interaction. Validate Active Directory for backdoor accounts, delegated permissions, and Kerberoastable SPNs created during the intrusion window before declaring full recovery.
Forensic Artifacts	Exchange IIS logs (C:\inetpub\logs\LogFiles\W3SVC1\) — EWS POST requests to /ews/exchange.asmx from internal hosts not operating mail clients; user-agent strings and authentication context identify Kazuar's EWS-based C2 channel (T1071.003) | Sysmon Event IDs 17 and 18 (Pipe Created / Pipe Connected) — named pipe events from non-mail processes on Windows endpoints; Kazuar uses named pipes for intra-botnet P2P communication between Tier-2 and Worker nodes | Windows Registry HKLM\SOFTWARE\Classes\CLSID and HKCU\SOFTWARE\Classes\CLSID — diff against clean baseline to identify COM object registrations or hijacks introduced by Kazuar's .NET loader (T1546.015) | Full volatile memory image from suspected Kernel and Worker nodes — Kazuar stages payloads in-memory via its .NET loader; RAM capture (WinPmem or Magnet RAM Capture) is required to recover implant code that may leave no persistent on-disk artifact | Windows Security Event Log Event IDs 4698 and 4702 (Scheduled Task Created/Updated) and Autoruns export from C:\Windows\System32\Tasks\ — Kazuar establishes autostart persistence (T1547) via scheduled tasks invoking .NET assemblies or encoded payloads, leaving task XML artifacts as primary on-disk persistence evidence

Per-Action IR Details

Containment — Audit Microsoft Exchange environments for anomalous EWS activity immediately. Block or restrict EWS access to known, authorized clients using Exchange throttling policies and network controls. Isolate any Windows hosts exhibiting unexplained named pipe or mailslot activity pending investigation.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST IR-4 (Incident Handling), NIST AC-4 (Information Flow Enforcement), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 13.4 (Perform Traffic Filtering Between Network Segments)

Compensating: On Exchange, run Get-ThrottlingPolicy and review EWS-specific throttling (EWSMaxConcurrency, EWSPercentTimeInCAS). Create a dedicated throttling policy with EWSEnabled $false applied to service accounts not requiring EWS: New-ThrottlingPolicy -Name 'BlockEWS' -EWSEnabled $false; Set-Mailbox -Identity -ThrottlingPolicy BlockEWS. For named pipe isolation on Windows hosts, use Sysinternals PipeList.exe to enumerate active named pipes and compare against a known-good baseline from a clean reference system. Isolate suspect hosts by disabling their NIC via PowerShell: Disable-NetAdapter -Name 'Ethernet' -Confirm:$false.

Evidence: Before isolating, capture: (1) Exchange IIS logs from C:\inetpub\logs\LogFiles\W3SVC1\ — extract all EWS POST requests with client IP, user-agent, and authenticated user for the prior 90 days; (2) Windows Security Event Log Event ID 4624/4625 on Exchange server for logon events correlating to anomalous EWS sessions; (3) Sysinternals PipeList or Sysmon Event ID 17 (Pipe Created) logs showing named pipe names — Kazuar uses named pipes for intra-node P2P communication; (4) Mailslot activity via Procmon filter on 'Operation is CreateFile' with Path contains '\\.\mailslot' to capture Kazuar's inter-node messaging artifacts before host isolation.

Detection — Query Exchange transport logs and IIS logs for atypical EWS client patterns, especially repeated programmatic access from internal hosts that are not mail clients. Use Sysmon Event IDs 17 and 18 to detect named pipe creation and connection events. Hunt for .NET loader artifacts and COM object registration

changes on endpoints in government, diplomatic, or critical infrastructure segments. Correlate with MITRE T1071.003 and T1572 behavioral signatures in your SIEM.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: For EWS hunting without SIEM: use PowerShell to parse IIS logs — Select-String -Path 'C:\inetpub\logs\LogFiles\W3SVC1*.log' -Pattern '/ews/exchange.asmx' | Where-Object {$_ -match 'POST'} | Export-Csv ews_hits.csv. For Sysmon pipe detection, deploy Sysmon with SwiftOnSecurity config and filter for Event ID 17 where PipeName matches known Kazuar pipe naming patterns (random or GUID-formatted names spawned by non-mail processes). For COM object hunting, run reg query HKLM\SOFTWARE\Classes\CLSID /s and diff against a baseline snapshot — Kazuar's .NET loader may register or hijack COM objects (T1546.015). Use Sigma rule 'proc_creation_win_dotnet_clr_unusual_process' adapted for Exchange service context.

Evidence: Before completing detection sweep, preserve: (1) Exchange EWS application pool logs at C:\Windows\System32\LogFiles\HTTPERR\ for connection errors indicating probing or automation; (2) Sysmon Event ID 7 (Image Loaded) logs showing .NET CLR DLLs (clr.dll, mscorlib.dll) loaded into non-.NET-native processes — Kazuar's loader stages .NET execution inside unexpected host processes; (3) Registry snapshot of HKLM\SOFTWARE\Classes\CLSID and HKCU\SOFTWARE\Classes\CLSID for COM registration changes within the investigation window; (4) Windows Event ID 4688 (Process Creation) with full command-line logging enabled, filtering on parent processes being Exchange worker processes (w3wp.exe) spawning unusual children; (5) Network capture (Wireshark/tcpdump) on Exchange server NIC filtering for EWS SOAP envelope patterns from internal non-mail-client hosts.

Eradication — There is no patch to apply; this is implant-based compromise, not a patchable vulnerability. Eradication requires full forensic investigation of affected hosts, removal of the Kazuar implant and its persistence mechanisms (autostart entries per T1547), and revocation of any credentials present on compromised systems. Engage Microsoft Incident Response or a qualified DFIR team for confirmed infections.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST IA-4 (Identifier Management), CIS 5.3 (Disable Dormant Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: For persistence hunting without EDR: run Autoruns.exe (Sysinternals) with VirusTotal integration enabled — Kazuar persists via scheduled tasks, Run keys, and service registration (T1547). Export Autoruns results to CSV and diff against clean baseline: autorunsc.exe -a * -c -h -s > autoruns_output.csv. For credential revocation on compromised hosts, run Mimikatz in read-only mode on forensic copy to enumerate cached credentials before resetting — reset all domain accounts whose NTLM hashes were accessible on confirmed Kazuar nodes. Use YARA rule targeting Kazuar's known .NET loader signatures (published in Microsoft's Kazuar disclosure) against all files in %TEMP%, %APPDATA%, and Exchange installation directories.

Evidence: Before eradication actions, collect and preserve: (1) Full memory image using WinPmem or Magnet RAM Capture — Kazuar operates in-memory and may leave no on-disk implant on Worker nodes; (2) Scheduled task XML exports from C:\Windows\System32\Tasks\ for all tasks created or modified within the intrusion window, specifically tasks invoking .NET assemblies or encoded PowerShell; (3) Windows Security Event ID 4698 (Scheduled Task Created) and 4702 (Scheduled Task Updated) from the Security Event Log; (4) Service registry keys at HKLM\SYSTEM\CurrentControlSet\Services\ snapshotted for Kazuar-installed services; (5) All user profile %APPDATA% and %LOCALAPPDATA% directories on confirmed compromised hosts for .NET assembly artifacts (.dll, .exe) with recent creation timestamps matching the intrusion window.

Recovery — Validate EWS access controls post-remediation and confirm named pipe and mailslot activity has returned to baseline. Re-image confirmed Kernel and Worker nodes rather than attempting in-place cleanup. Monitor Exchange EWS logs and endpoint telemetry for at least 30 days post-remediation given the botnet's leader election capability, which allows surviving nodes to reconstitute C2.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CP-10 (System Recovery and Reconstitution), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Validate EWS baseline post-remediation by re-running the IIS log PowerShell query from detection phase and confirming only authorized mail clients appear. Use PipeList.exe on rebuilt hosts and compare output against the clean reference baseline captured during containment. For 30-day monitoring without SIEM, configure Windows Event Forwarding (WEF) to centralize Sysmon Event IDs 17, 18, and 7 from recovered hosts to a dedicated collector, then run daily PowerShell parsing: Get-WinEvent -ComputerName -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational'; Id=17} | Where-Object {$_.TimeCreated -gt (Get-Date).AddDays(-1)}.

Evidence: Before returning systems to production, verify: (1) Re-image validation via file integrity check — use FCIV or certutil -hashfile on OS binaries against known-good hashes from Microsoft's file hash database; (2) Confirm Exchange EWS application pool identity accounts have had passwords rotated and service principal names (SPNs) re-registered; (3) Validate no new COM CLSID registrations exist on rebuilt hosts beyond the clean baseline; (4) Confirm Active Directory for any new accounts, group membership changes, or Kerberoastable service accounts created during the intrusion window using AD audit logs (Event IDs 4720, 4728, 4732, 4756); (5) Run a final Autoruns diff on rebuilt hosts to confirm persistence mechanisms are absent before production return.

Post-Incident — Review whether behavioral detection rules for protocol tunneling (T1572) and internal proxy use (T1090.001) exist and are tuned. Assess whether EWS is required in your environment; disable it if not. Evaluate whether endpoint telemetry (Sysmon or equivalent) covers named pipe and COM object activity. This campaign exposes gaps in monitoring legitimate protocol channels for C2 abuse.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-4 (System Monitoring), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure)

Compensating: To disable EWS on Exchange if not required: Set-OrganizationConfig -EwsEnabled $false, or scope to specific users via throttling policy. Deploy the Sigma rule 'win_susp_named_pipe_client' and 'win_mal_kazuar' (if published) to your log pipeline — for teams without SIEM, translate Sigma rules to PowerShell Event Log queries using the sigma-cli converter targeting Windows Event Log backend. Submit Kazuar .NET loader YARA signatures to ClamAV for ongoing endpoint scanning. Document EWS as a C2-capable protocol in your threat model and add it to quarterly access control reviews per CIS 7.2 (Establish and Maintain a Remediation Process).

Evidence: For lessons-learned documentation, preserve the following as evidence of detection gaps: (1) The EWS IIS log gap period — the time window between first anomalous EWS access and detection, demonstrating dwell time; (2) Sysmon configuration file in use at time of compromise, to evidence whether Event ID 17/18 pipe monitoring was enabled or absent; (3) AD audit log coverage gap analysis — document which Event IDs were and were not being collected at time of intrusion; (4) Network flow records (NetFlow or firewall logs) showing internal host-to-Exchange EWS communication patterns that were not alerted on, to justify new detection rule development; (5) Timeline of Kazuar node roles (Kernel, Tier-2, Worker) identified during investigation, mapped to internal IP addresses, to evidence the blast radius and inform network segmentation improvements.

Detection Guidance

Primary detection focus is on EWS abuse and intra-host communication primitives. In Exchange logs (EWS logs at %ExchangeInstallPath%\Logging\EWS), look for high-frequency or scripted access patterns from internal non-mail-client hosts, particularly those making repeated item fetch or folder-sync calls. In Sysmon logs, Event ID 17 (Pipe Created) and Event ID 18 (Pipe Connected) should be reviewed for pipes with non-standard or randomized names. COM object registration changes can be detected via registry monitoring on HKLM\Software\Classes\CLSID. Process injection indicators (T1055) include unexpected cross-process memory writes visible in Sysmon Event ID 8 (CreateRemoteThread). For network-layer detection, inspect TLS-encrypted outbound traffic for protocol tunneling patterns (T1572); JA3/JA3S fingerprinting may assist. Behavioral rules mapped to T1090.001 (internal proxy) should flag hosts forwarding traffic on behalf of other internal nodes. Microsoft Defender for Endpoint customers should review alerts mapped to the Secret Blizzard threat actor cluster. Threat hunting hypothesis: identify Windows hosts that have established named pipe connections to multiple other internal hosts within a short window, cross-referenced against hosts with anomalous EWS access.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`EWS endpoint abuse pattern — programmatic access via Exchange Web Services from non-mail-client internal hosts`	Kazuar uses EWS as a C2 channel; specific IOC values not published in available source material	MEDIUM
DOMAIN	`[not published in available source]`	Microsoft's analysis references C2 infrastructure but specific domains were not extracted from the available source tier; refer to the full Microsoft threat intelligence report for published IOCs	LOW

Framework Mappings

MITRE-ATTACK

- **T1071.001** — Web Protocols
- **T1041** — Exfiltration Over C2 Channel
- **T1572** — Protocol Tunneling
- **T1056.001** — Keylogging
- **T1567** — Exfiltration Over Web Service
- **T1021.002** — SMB/Windows Admin Shares
- **T1036** — Masquerading
- **T1083** — File and Directory Discovery
- **T1090.001** — Internal Proxy
- **T1071.003** — Mail Protocols
- **T1055** — Process Injection

- **T1573.001** — Symmetric Cryptography
- **T1113** — Screen Capture
- **T1560** — Archive Collected Data
- **T1059** — Command and Scripting Interpreter
- **T1105** — Ingress Tool Transfer
- **T1547** — Boot or Logon Autostart Execution
- **T1095** — Non-Application Layer Protocol
- **T1497** — Virtualization/Sandbox Evasion
- **T1059.001** — PowerShell

NIST-800-53R5

- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-4** — System Monitoring
- **AC-6** — Least Privilege
- **SI-3** — Malicious Code Protection
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1071.001	Web Protocols	Command-And-Control

Technique ID	Technique Name	Tactic
T1041	Exfiltration Over C2 Channel	Exfiltration
T1572	Protocol Tunneling	Command-And-Control
T1056.001	Keylogging	Collection
T1567	Exfiltration Over Web Service	Exfiltration
T1021.002	SMB/Windows Admin Shares	Lateral-Movement
T1036	Masquerading	Defense-Evasion
T1083	File and Directory Discovery	Discovery
T1090.001	Internal Proxy	Command-And-Control
T1071.003	Mail Protocols	Command-And-Control
T1055	Process Injection	Defense-Evasion
T1573.001	Symmetric Cryptography	Command-And-Control
T1113	Screen Capture	Collection
T1560	Archive Collected Data	Collection
T1059	Command and Scripting Interpreter	Execution
T1105	Ingress Tool Transfer	Command-And-Control
T1547	Boot or Logon Autostart Execution	Persistence
T1095	Non-Application Layer Protocol	Command-And-Control
T1497	Virtualization/Sandbox Evasion	Defense-Evasion
T1059.001	PowerShell	Execution

Sources

Source	URL	Tier
Microsoft Security Blog	https://www.microsoft.com/en-us/security/blog/2026/05/14/kazuar-ana...	T1
	https://www.microsoft.com/en-us/security/blog/2026/05/14/kazuar-ana...	T1
Russian state hackers exploit new Microsoft Office flaw in attacks on ...	https://therecord.media/russian-state-hackers-exploit-new-microsoft...	T3

Source	URL	Tier
Microsoft discloses malware attack on Ukraine govt networks	https://apnews.com/article/technology-business-europe-russia-ukrain...	T2
Windows zero-day actively exploited to spy on European diplomats	https://www.bleepingcomputer.com/news/security/chinese-hackers-expl...	T3

Generated 2026-05-14 13:49 UTC by TJS Security Command Center