INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:36 UTC

GSocket Weaponized as Covert Backdoor: Bash-Delivered Tunnel Bypasses Firewalls and NAT

THREAT CAMPAIGN | **HIGH** | CVSS 5.0

SCC Item ID	SCC-CAM-2026-0050
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	5.0
Affected Products	Linux/Unix systems (Ubuntu confirmed); any environment where GSocket (hackerschoice/gsocket) can be installed via bash script
Published	2026-03-21

Executive Summary

Threat actors are abusing GSocket, a legitimate open-source tunneling tool, to establish persistent encrypted backdoors on Linux systems by delivering it through malicious bash scripts. Because GSocket creates outbound-only encrypted tunnels, traditional perimeter firewalls and NAT controls provide no effective defense, leaving affected organizations with unauthorized remote access that is difficult to detect at the network edge without endpoint telemetry. Any Linux environment where an attacker can execute a bash script is at risk, and the primary business impact is covert, long-term access to internal systems without triggering standard network-based alerting.

Technical Analysis

This is a tool-abuse campaign with no associated CVE. Attackers deliver GSocket (hackerschoice/gsocket) via bash-based installers or droppers that download and execute the tool on Linux/Unix hosts, with Ubuntu confirmed affected. GSocket uses SRP (Secure Remote Password) authentication and AES-256-CBC-SHA end-to-end encryption over a shared secret, making tunnel traffic indistinguishable from legitimate encrypted communications at the network layer. The tunnel requires only outbound connectivity, bypassing firewall ingress rules and NAT entirely. Persistence is established via systemd service creation (T1543.003). The full MITRE ATT&CK technique set includes: T1105 (Ingress Tool Transfer), T1027 (Obfuscated Files or Information), T1059.004 (Unix Shell), T1021.004 (SSH), T1543.003 (Systemd Service), T1090.003 (Multi-hop Proxy), T1572 (Protocol Tunneling), T1071.001 and T1071.002 (Application Layer Protocol). Applicable CWEs are CWE-506 (Embedded Malicious Code) for the weaponized delivery mechanism and CWE-284 (Improper Access Control)

for the resultant unauthorized tunnel. No vendor patch is applicable; GSocket itself is not vulnerable software. Mitigation requires process-level and behavioral controls, not network patching. Primary sources are the GSocket GitHub repository and community red-team documentation. Independent threat research corroboration is recommended before escalating organizational response.

Action Checklist

1. Step 1, Immediate: Search all Linux endpoints for the GSocket binary (gs-netcat, gs-sftp, gsocket), the default install path (~/.gsocket or /usr/local/bin/gs-netcat), and any running processes matching those names. Terminate and quarantine confirmed instances.
2. Step 2, Detection: Query EDR and process telemetry for bash script executions that download and pipe to shell (curl | bash, wget | bash patterns), followed by gs-netcat or gsocket process spawns. Review systemd unit files created within the last 90 days for unfamiliar service definitions referencing gs-netcat or gsocket binaries.
3. Step 3, Assessment: Inventory all Linux/Unix systems for presence of the hackerschoice/gsocket package or binary artifacts. Cross-reference against authorized software lists. Flag any host where GSocket is present but was not explicitly approved and deployed by IT or security teams.
4. Step 4, Egress Review: Audit outbound network connections from Linux hosts to identify sustained low-bandwidth encrypted sessions on non-standard ports or to unfamiliar external IPs, particularly those with no corresponding inbound traffic. GSocket relays through externally-controlled infrastructure (by default, the HackerChoice relay service), making direct C2 IP identification at the perimeter unreliable; focus detection on endpoint behavior.
5. Step 5, Long-term: Update Linux endpoint security baselines to explicitly prohibit installation of unapproved tunneling tools. Add GSocket binary hashes and process names to EDR block lists. Implement or tune behavioral detection rules for curl-to-shell and wget-to-shell execution chains. Review and restrict which accounts can install software or create systemd services on production Linux hosts.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO/IR leadership if GSocket is found on more than 5% of Linux fleet, if C2 egress is confirmed to external infrastructure, or if evidence indicates persistence mechanism (systemd service or cron job) suggesting active attacker presence rather than isolated testing.
Recovery Notes	After eradication: (1) Rebuild affected systems from clean base image or apply hardened Linux security baseline (CIS Benchmarks) and redeploy approved applications only. (2) Rotate SSH keys, service account credentials, and API tokens on all compromised hosts and dependent systems, particularly any systems those accounts accessed during the infection window. (3) Audit all outbound firewall rules and proxy logs for the 90 days prior to detection to identify other potential C2 beacons or lateral movement, and expand EDR deployment to catch future abuse.

Forensic Artifacts	/var/log/auth.log and /var/log/auth.log.* (sudo/su command history, package manager activity) | /root/.bash_history, /home/*/.bash_history, ~/.zsh_history (user command history with curl|bash, wget|bash patterns) | systemd journal: journalctl --since '90 days ago' (service creation, process spawning, unit load events) | /etc/systemd/system/*.service and /etc/systemd/user/*.service (persistence mechanisms, last-modified timestamps) | Package manager logs: /var/log/apt/history.log*, /var/log/yum.log*, /var/log/zypper.log* (installation timeline) | Process accounting: /var/log/account/pacct (if enabled, full process execution history with exit codes) | File system inode metadata: stat output for any gs-netcat binary and parent directory timestamps (installation date confirmation) | Network connection logs: netstat/ss snapshots, packet captures on non-standard egress ports, firewall logs from host-based UFW/firewalld | DNS query logs (if available): /var/log/syslog* grep for DNS lookups to external relay infrastructure during suspected infection window

Per-Action IR Details

Step 1 — Immediate: Search all Linux endpoints for the GSocket binary (gs-netcat, gs-sftp, gsocket), the default install path (~/.gsocket or /usr/local/bin/gs-netcat), and any running processes matching those names. Terminate and quarantine confirmed instances.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2.3 (Containment Strategy)

Controls: NIST SI-7 (Software, Firmware, and Information Integrity Monitoring), CIS 6.2 (Actively manage all software on network devices)

Compensating: Execute `find / -name 'gs-netcat' -o -name 'gsocket' -o -name 'gs-sftp' 2>/dev/null` across all Linux hosts via SSH loop or Ansible ad-hoc command. Cross-reference against `ps aux | grep gs-` for running processes. Capture full process details with `ps -ef | grep gs-netcat` before killing. Use `kill -9 ` to terminate, then move binaries to isolated forensic hold directory (/var/quarantine/gsocket-evidence-YYYYMMDD/) with permissions 000.

Evidence: Capture before termination: (1) Full process listing `ps -auxww | grep gs-netcat > /tmp/gsocket-processes-PRE.txt`, (2) Open file descriptors `lsof -p ` for each gs-netcat process to identify C2 destinations, (3) Binary file metadata `ls -laRi /usr/local/bin/gs-netcat ~/.gsocket 2>/dev/null > /tmp/gsocket-inode-map.txt`, (4) Recent modification times via `stat ` and inode change times.

Step 2 — Detection: Query EDR and process telemetry for bash script executions that download and pipe to shell (curl | bash, wget | bash patterns), followed by gs-netcat or gsocket process spawns. Review systemd unit files created within the last 90 days for unfamiliar service definitions referencing gs-netcat or gsocket binaries.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.1 (Detection and Analysis)

Controls: NIST SI-4 (Information System Monitoring), CIS 8.5 (Log all access to audit records), CIS 6.4 (Implement and test an automated software inventory tool)

Compensating: Without EDR: (1) Query bash command history across all users: `find / -name '.bash_history' -o -name '.zsh_history' 2>/dev/null | xargs grep -l 'curl.*|.*bash\|wget.*|.*bash' 2>/dev/null`, (2) Check systemd service creation logs: `journalctl --since '90 days ago' | grep -i 'unit.*created\|gsocket'` and `ls -lat /etc/systemd/system/*.service /etc/systemd/user/*.service 2>/dev/null | head -20`, (3) Audit process accounting if enabled: `lastcomm | grep -E 'curl|wget|bash' | tail -100`, (4) Parse auth.log for sudo-invoked installations: `grep 'COMMAND=.*curl\|COMMAND=.*wget' /var/log/auth.log*`.

Evidence: Preserve before querying: (1) Complete bash history files from all user home directories (especially root, service accounts, web app users), (2) Entire /etc/systemd/system/ and /etc/systemd/user/ directories with timestamps, (3) Full /var/log/auth.log* (all rotated copies within 90 days), (4) /var/log/syslog or /var/log/messages for package manager activity, (5) Systemd journal: `journalctl --since '90 days ago' > /tmp/journal-dump-90d.txt`.

Step 3 — Assessment: Inventory all Linux/Unix systems for presence of the hackerschoice/gsocket package or binary artifacts. Cross-reference against authorized software lists. Flag any host where GSocket is present but was not explicitly approved and deployed by IT or security teams.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.1 (Detection and Analysis) and NIST 800-53 CM-8 (Information System Component Inventory)

Controls: NIST CM-8 (Information System Component Inventory), CIS 2.1 (Maintain inventory of all hardware devices and software), CIS 6.4 (Implement software inventory)

Compensating: Script a parallel inventory across all hosts using SSH: `for host in $(cat /tmp/linux-hosts.txt); do ssh $host 'echo "=== $host ===" && find / -name '*gsocket*' 2>/dev/null && dpkg -l 2>/dev/null | grep -i gsocket && rpm -qa 2>/dev/null | grep -i gsocket' >> /tmp/gsocket-inventory.log; done`. Compare output against change control documentation and approved software baseline. Flag any host with GSocket presence not traceable to a signed change request with IT approval and deployment date.

Evidence: Retain: (1) Complete output log from inventory scan with timestamps, (2) Change control records for any system claiming authorized GSocket deployment, (3) Package manager logs showing installation date: `grep -i gsocket /var/log/apt/history.log* /var/log/yum.log* /var/log/zypper.log* 2>/dev/null`, (4) Filesystem metadata for any installed GSocket artifact: `stat -c '%n %y %a %U:%G' /usr/local/bin/gs-netcat 2>/dev/null`.

Step 4 — Egress Review: Audit outbound network connections from Linux hosts to identify sustained low-bandwidth encrypted sessions on non-standard ports or to unfamiliar external IPs, particularly those with no corresponding inbound traffic. GSocket relays through external infrastructure, so direct C2 IP identification at the perimeter is unreliable.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.2 (Analysis) and NIST 800-53 SI-4(4) (Automated Monitoring for Unusual Activity)

Controls: NIST SI-4 (Information System Monitoring), CIS 9.3 (Block inbound traffic by default)

Compensating: Compensating without NetFlow/proxy: (1) Export 30-day netstat snapshots from each Linux host: `netstat -tulpn 2>/dev/null | grep ESTABLISHED > /tmp/netstat-baseline-$HOSTNAME.txt && ss -tulpn 2>/dev/null | grep ESTABLISHED >> /tmp/netstat-baseline-$HOSTNAME.txt`, (2) Collect packet captures on suspected hosts: `tcpdump -i any -w /tmp/egress-capture-$HOSTNAME.pcap 'src host and dst port not (22|53|80|443|123|587|3306)' -G 3600 -W 24` (24-hour rolling capture), (3) Parse /var/log/syslog for connection attempts and extract destination IPs: `grep -oE '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+' /var/log/syslog* | sort | uniq -c | sort -rn`, (4) If auditd is enabled, query outbound connect syscalls: `ausearch -m SOCKADDR --raw | grep -v 127.0.0.1 | tail -500`.

Evidence: Capture immediately: (1) Live netstat/ss output showing all ESTABLISHED connections with associated PIDs and owner usernames, (2) 30-day historical network logs from any host-based firewall (ufw, firewalld, iptables rules), (3) Continuous packet capture on suspected hosts focused on non-standard egress ports, (4) Process-to-network mapping: `netstat -tulpn 2>/dev/null | awk '{print $NF}' | cut -d/ -f2 | sort -u | xargs -I {} ps -p {} -o pid,user,comm,args` to link PIDs to commands, (5) DNS query logs if available: `grep 'query' /var/log/syslog* 2>/dev/null | grep -oE '[a-zA-Z0-9.-]+\.(com|net|org|io)' | sort | uniq > /tmp/dns-queries-30d.txt`.

Step 5 — Long-term: Update Linux endpoint security baselines to explicitly prohibit installation of unapproved tunneling tools. Add GSocket binary hashes and process names to EDR block lists. Implement or tune behavioral detection rules for curl-to-shell and wget-to-shell execution chains. Review and restrict which accounts can install software or create systemd services on production Linux hosts.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §3.2.5 (Post-Incident Activities) and NIST 800-53 CM-6 (Configuration Settings)

Controls: NIST AC-2 (Account Management and AC-3 Access Enforcement), NIST SI-7 (Software Integrity Monitoring), NIST SI-2 (Flaw Remediation), CIS 5.3 (Configure sudo to log all commands and results), CIS 6.2 (Ensure all software is authorized)

Compensating: Without EDR block lists: (1) Generate binary hash signatures via `sha256sum /path/to/gs-netcat >> /etc/gsocket-ioc-hashes.txt` and deploy AppArmor or SELinux policy to prevent execution: `echo

'/usr/local/bin/gs-netcat flags=(attach_disconnected) { deny / rwk, }' >> /etc/apparmor.d/usr.local.bin.gs-netcat` then `apparmor_parser -r /etc/apparmor.d/usr.local.bin.gs-netcat`, (2) Restrict sudo package installation via sudoers: `echo 'Defaults log_output, logfile="/var/log/sudo-cmds.log"' >> /etc/sudoers` and `echo '%sudo ALL=(ALL) !apt-get install, !apt install, !pip install' >> /etc/sudoers`, (3) Create auditd rules to alert on pipe-to-shell chains: `auditctl -a always,exit -F arch=b64 -S execve -F exe=/usr/bin/bash -F a1='-c' -k shell-injection`, (4) Implement file integrity monitoring: `aide --init && mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz` and schedule hourly checks, (5) Restrict write access to /usr/local/bin and /etc/systemd/system to root only with no sudo delegation for application service accounts.

Evidence: Document baseline: (1) Approved software whitelist with authorized version numbers and deployment dates, (2) Baseline AppArmor/SELinux policy files with signatures, (3) Sudoers baseline configuration with audit rules, (4) AIDE/tripwire baseline hash database, (5) Approved systemd service unit templates for reference, (6) Change control process documentation showing review and approval workflow for all future Linux privilege escalation or software installation requests.

Detection Guidance

Detection must rely on endpoint and process telemetry, not network-layer controls. Key behavioral indicators: (1) Bash or sh processes executing curl or wget commands that pipe directly to bash or sh, particularly with short-lived parent processes. (2) Creation of new systemd unit files in /etc/systemd/system/ or ~/.config/systemd/user/ referencing binaries in non-standard paths. (3) Presence of gs-netcat, gsocket, or gs-sftp binaries anywhere on the filesystem, run: find / -name 'gs-netcat' -o -name 'gsocket' 2>/dev/null. (4) Outbound TCP or UDP connections from Linux hosts that sustain encrypted sessions with no corresponding inbound session and no process attributed to a known application. (5) Processes spawned with environment variables or arguments referencing a GSocket secret key (-s flag). SIEM query pattern (Syslog/Auditd): alert on execve events where argv contains 'gs-netcat' or 'gsocket'. EDR-based: flag any process whose binary path resolves to a GSocket binary hash. Note: no confirmed IOC hashes, IPs, or domains are available from current sources. The indicators above are behavioral, not signature-based. If campaign-specific IOCs are published, they should be integrated into SIEM and EDR detection rules immediately. Independent threat research should be consulted for campaign-specific IOCs before tuning detection rules.

Indicators of Compromise

Type	Value	Context	Confidence
HASH	`not available`	No confirmed malicious GSocket binary hashes have been published in available sources. Monitor for the legitimate GSocket binary used in an unauthorized context rather than relying on hash-based detection.	LOW
DOMAIN	`gsocket.io`	Legitimate GSocket relay infrastructure domain. Unexpected outbound connections to this domain from production Linux hosts that do not have authorized GSocket deployments warrant investigation. This is not inherently malicious — context and authorization status determine risk.	MEDIUM

Type	Value	Context	Confidence
URL	`https://github.com/hackerschoice/gsocket`	Official GSocket repository. Presence of download activity from this URL in proxy or DNS logs on non-developer hosts may indicate installation activity.	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1105** — Ingress Tool Transfer
- **T1027** — Obfuscated Files or Information
- **T1059.004** — Unix Shell
- **T1021.004** — SSH
- **T1543.003** — Windows Service
- **T1090.003** — Multi-hop Proxy
- **T1572** — Protocol Tunneling
- **T1071.002** — File Transfer Protocols
- **T1071.001** — Web Protocols

NIST-800-53R5

- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **AC-3** — Access Enforcement
- **AT-2** — Literacy Training and Awareness
- **SC-13** — Cryptographic Protection

OWASP-TOP10-2021

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

CIS-V8

- **6.1**
- **6.2**
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **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
- **164.312(e)(1)** — Transmission Security

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.8.24** — Use of cryptography

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1105	Ingress Tool Transfer	Command-And-Control
T1027	Obfuscated Files or Information	Defense-Evasion
T1059.004	Unix Shell	Execution
T1021.004	SSH	Lateral-Movement
T1543.003	Windows Service	Persistence
T1090.003	Multi-hop Proxy	Command-And-Control
T1572	Protocol Tunneling	Command-And-Control
T1071.002	File Transfer Protocols	Command-And-Control
T1071.001	Web Protocols	Command-And-Control

Sources

Source	URL	Tier
Security News	https://www.gsocket.io/	T3
hackerschoice/gsocket: Connect like there is no firewall. Securely.	https://github.com/hackerschoice/gsocket	T3
Security Overview · hackerschoice/gsocket - GitHub	https://github.com/hackerschoice/gsocket/security	T3

Source	URL	Tier
Issues · hackerschoice/gsocket - GitHub	https://github.com/hackerschoice/gsocket/issues	T3
GSocket for Persistence | Infiltr8 - The Red-Book	https://red.infiltr8.io/redteam/persistence/linux/gsocket-for-persi...	T3

Generated 2026-03-29 18:36 UTC by TJS Security Command Center