## Introduction

VerSprite was asked to conduct a Source Code and Design Review on behalf of Norton VPN. The test took place between August 28th, 2025, and September 26th, 2025, with the consent and full knowledge of Norton VPN officials. Before conducting the Source Code and Design Review, a formal kick-off email was sent to ensure that all members, from both VerSprite and Norton VPN, were adequately informed of the risks, level of effort, points of contact, and expected duration of the assessment. A validation retest was performed on November 11th, 2025, and this report was updated accordingly.

## Conclusions

Considering that all issues presented in this report have been fixed in the latest version of the code at the time of writing, the overall technical risk for ***Mimic Protocol Security Testing*** based on the Source Code and Design Review and the impact of discovered vulnerabilities is ***None***. This score takes into consideration the lack of open Critical, High, Medium, and Low-Risk vulnerabilities across all phases of the Source Code and Design Review. Furthermore, the score reflects the likelihood of exploitation, existing threats, and the overall business impact based on VerSprite's assessment of the criticality of the assets and data at risk.

## Objectives and Scope

The primary objective for this Source Code and Design Review was to identify high-impact vulnerabilities within the Mimic Protocol Security Testing, which could lead to exploitation, theft of confidential user data, and overall privilege escalation. The Source Code and Design Review followed a method intended to simulate real-world attack scenarios and threats that could critically impact data privacy, authenticity, integrity, and overall business reputation.

## Methodology

VerSprite primarily performed manual source code review but added automated testing for a breadth of coverage or, when necessary, to complement specific tests. Additionally, VerSprite compiled the provided source code and set up a virtual test environment, where any potential vulnerability detected in the source code would be confirmed by runtime tests. VerSprite also performed some runtime-only tests to discover potential vulnerabilities that would be difficult to spot using only static code analysis, but would stand out in a live environment.

## Overall Findings and Remediation Notes

During the security exercise, we identified a possible ***Nloc Authentication HTTP Parameter Pollution (VSID-001)*** vulnerability. Under certain Mimic server configuration settings, a local daemon is called using an HTTP API to validate the user credentials. We discovered the code that constructed the URL to call this API was theoretically vulnerable to parameter injection. Although we could not validate this vulnerability was actually exploitable in a realistic scenario, we opted to err on the side of caution and assume it was exploitable until proven otherwise. Norton VPN fixed this issue by refactoring the way the URL was built, using URL handling components rather than direct string manipulation, thus avoiding parameter injections entirely.

We also discovered a ***Certificate Authority with No Validation (VSID-002)***. The Tenta API, used by a configuration script within the source code, allows the creation of valid TLS certificates for any host that would, in principle, be accepted by a Mimic client. This opened up the possibility of carrying out Man-in-the-Middle attacks, since an attacker could easily impersonate any Mimic server that uses this CA by just requesting a new certificate for it. This CA does not appear to be mandatory, and our research found multiple servers in the wild that do not use it – nevertheless, we would have been remiss not to point out the existence of this PKI mechanism that is insecure by design. Norton VPN responded to this issue by locking down the Tenta API CA, making it inaccessible to the internet, while keeping it available internally for testing

purposes. This also mitigates the issue for existing servers using this CA, since attackers can no longer generate new certificates.

Regarding the stealth properties of this VPN protocol, we discovered two ways in which it could be broken. In issue ***Mimic Protocol ALPN Disclosure (VSID-003)*** we discussed how the Mimic protocol negotiated its version via the Application-Layer Protocol Negotiation feature of TLS, which happens before the encrypted channel is set up (specifically in the Client Hello packet and its response). Since this occurs before encryption itself, it is vulnerable to passive surveillance. A malicious actor with the capability to intercept (but not decrypt) network traffic would be able to identify connections to a Mimic VPN server which, under some scenarios, can be a problem. (Consider for example users behind restricted environments such as the Chinese Firewall, where any detected VPN connection may be a problem by itself). Norton VPN addressed this issue by creating a new version of the Mimic protocol (version 7) where negotiation occurs within the protocol itself, rather than using ALPN data. While older clients and servers remain vulnerable (fixing them without breaking compatibility would be impossible), updated clients and servers will benefit from this security fix.

Furthermore, in issue ***Mimic Protocol HTTP Disclosure (VSID-005)*** we discuss how Mimic servers could be trivially discovered online via a port scan or a simple HTTP request. We further illustrated the problem by showing over 1000 available Mimic servers that could be discovered via the Shodan search engine at the time of writing. Ideally the Mimic servers should be mostly indistinguishable from a regular HTTPS server to prevent actors from discovering VPN connections simply by monitoring access to known Mimic servers. Norton VPN addressed this issue in a new version of the daemon that no longer responds to HTTP requests in a way that can identify the service as a Mimic service.

Additionally, we found a ***Potential Denial of Service (SlowLoris) (VSID-004)*** affecting both the main Mimic port and a performance monitoring port, with the first being the main focus of our security concerns. A step-by-step description of the locations in the source code that make this possible was provided, and we carried out a simulation of such an attack against a test server. Norton VPN fixed this issue by adding timeouts to all network operations, exactly as recommended and following best practices.

Some low-risk issues were also identified. While low risk issues may not constitute dangerous vulnerabilities by themselves, they may still affect the Mimic server by either providing information to attackers or by chaining them with other vulnerabilities. In particular, we found that it was possible to cause a ***Fallback to Weak TLS Algorithms (VSID-006)*** during a Man-in-the-Middle attack against a Mimic connection, in which the client was manipulated into connecting using a ciphersuite that did not provide quantum resistance or perfect forward secrecy. Additionally, the ***Use of Outdated Component with Known Vulnerabilities (VSID-007)*** was found as well, as the customized Go compiler used by Mimic, patched to provide quantum resistant ciphersuites, was downloading and patching version 1.24.1 of the vanilla Go compiler. This was one version behind at the time of writing – the updated version provides protection against a number of publicly disclosed vulnerabilities. Norton VPN fixed both issues by removing the weak TLS algorithms and upgrading the Go compiler to the latest version.