Mullvad VPN Platform Security Assessment

Security Assessment Report

MULLVAD VPN

Prepared for Mullvad VPN AB
October 14, 2022 (version 1.0)

Project Team:

Technical Testing	Jordan Whitehead and William Vu
Technical Editing	Nick Nam and Sara Bettes
Project Management	Sara Bettes

Table of Contents

Engagement Overview

Assessment Components and Objectives

Mullvad VPN AB ("Mullvad") recently engaged Atredis Partners ("Atredis") to perform a Security Assessment of the Mullvad VPN clients. Objectives included validation that the Mullvad VPN clients were developed and deployed with security best practices in mind, and to obtain third party validation that any significant vulnerabilities present in the Mullvad VPN clients were identified for remediation.

Testing was performed from September 6 through October 6, 2022, by Jordan Whitehead and William Vu of the Atredis Partners team, with Sara Bettes providing project management and delivery oversight. For Atredis Partners' assessment methodology, please see Appendix I of this document, and for team biographies, please see Appendix II. Specific testing components and testing tasks are included below.

COMPONENT	ENGAGEMENT TASKS
Mullvad VPN Platform Security Assessment	

Assessment Tasks	• Source-Assisted Runtime Assessment of Mullvad VPN clients
	  • Instrumented Security Review of Userland, Kernel Drivers
	  • Automated and runtime application testing
	  • Analysis of privilege model of Mullvad components
	  • Test case creation, fuzzing and fault injection
	  • PoC generation and validation of findings
	• Application API / web services penetration testing
	  • Application API automated and runtime testing
	  • Manual runtime analysis and protocol analysis
	  • Network attack chain review and traffic analysis

Reporting and Analysis	
Analysis and Deliverables	• Status Reporting and Realtime Communication • Comprehensive Engagement Deliverable • Engagement Outbrief and Remediation Review

The ultimate goal of the assessment was to provide a clear picture of risks, vulnerabilities, and exposures as they relate to accepted security best practices, such as those created by the National Institute of Standards and Technology (NIST), Open Web Application Security Project (OWASP), or the Center for Internet Security (CIS). Augmenting these, Atredis Partners also draws on its extensive experience in secure development and in testing high-criticality applications and advanced exploitation.

Engagement Tasks

Atredis Partners performed the following tasks, at a high level, for in-scope targets during the engagement.

Binary and Runtime Analysis

For relevant software targets identified during the course of this engagement, Atredis performed binary and runtime analysis, using debugging and instrumentation tools to analyze application flow to aid in software security analysis. Where relevant, purpose-built tools such as fuzzers and customized network clients may have been utilized to aid in vulnerability identification.

Configuration and Architecture Review

Atredis Partners performed a high-level review of available documentation and configuration data with an eye toward the overall functional design and soundness of the implementation. A key aspect of this component was identifying gaps in the architecture and design regarding aspects of design that reduce overall defensibility, aimed at pointing out fundamental issues in the application architecture that should be addressed early in the development cycle as opposed to later when the platform is closer to a full production state.

While specific vulnerabilities may be identified during the architecture and configuration review, the intent is less on finding individual defects and more on how the design of a given target affects its overall defensibility. Outcomes of the architecture review helped inform testing objectives throughout the rest of the engagement while also helping the client define a long-term platform maturity and security design roadmap.

Network Protocol Analysis

With the objective of identifying scenarios where the integrity of trusted communications can be diminished or reduced, Atredis Partners reviewed network traffic using various packet flow analysis and packet capture tools to observe in-scope network traffic. Network communications were analyzed for the presence of cleartext communications or scenarios where the integrity of cryptographic communications can be diminished, and Atredis attempted to identify means to bypass or circumvent network authentication or replay communications, as well as other case-dependent means to abuse the environment to disrupt, intercept, or otherwise negatively affect in-scope targets and communications.

Source Code Analysis

Atredis reviewed the in-scope application source code, with an eye for security-relevant software defects. To aid in vulnerability discovery, application components were mapped out and modeled until a thorough understanding of execution flow, code paths, and application design and architecture is obtained. To aid in this process, the assessment team engaged key stakeholders and members of the development team where possible to provide structured walkthroughs and interviews, helping the team rapidly gain an understanding of the application's design and development lifecycle.

Executive Summary

Atredis Partners was provided with five test accounts to use when testing the Mullvad clients on all the supported platforms. The `2022.4` tagged version of the Mullvad VPN application was analyzed for all the desktop clients. The Android client targeted was `android/2022.2-beta1` and the iOS client was `ios/2022.3-build1`.

Testing prioritized discovering possible traffic leaks and issues that could lead to the deanonymization of a user, or exposure of private data. The shared core logic was reviewed, along with the pieces unique to each platform.

In addition to identifying possible leak vectors, Atredis Partners also performed an assessment searching for vulnerabilities in the clients that could pose a security threat to users. This search included looking for possible attack vectors for a remote attacker, or a less privileged attacker on the same machine seeking to escalate to a higher privilege level.

The VPN clients are not designed to maintain their tunnel integrity in the event of a local attacker seeking to disable the VPN; such local attacks were not considered during the analysis.

The testing focus was on the Mullvad client applications and did not audit the security of the VPN servers or Mullvad API (Application Programming Interface) servers. Testing also did not directly audit the VPN providers used by the Mullvad clients (OpenVPN and well-known WireGuard implementations), except for any relevant changes made by Mullvad.

Key Conclusions

Overall, Atredis Partners found the Mullvad VPN clients to be well-architected from a security perspective, with limited attack surface that could be reached by an external malicious party, and important protection mechanisms were in place to prevent most unintended traffic leaks. Atredis Partners detected a few edge cases where traffic could be accidentally leaked outside the VPN tunnel. These leaks were either patched quickly by the Mullvad team or were due to the operating system itself, in which case the Mullvad team updated documentation and submitted issues to the operating system vendor where appropriate.

As in any security assessment, some areas for improvement were noted, but overall Atredis Partners would rate the Mullvad VPN clients as sound from a security perspective.

Architecture Analysis

The Mullvad VPN clients are built upon well-tested VPN providers such as OpenVPN and certain WireGuard implementations. The clients add value for a user by securely setting up the system to use these providers. The clients also introduce mechanisms for preventing traffic from accidentally not using the VPN and providing a "kill switch" to ensure that even in the event of some error, traffic will not unexpectedly start to go over the unsecured network.

Most of the clients share core logic from the `mullvad-daemon`, `talpid-core`, and associated code. This logic controls the state machine of the VPN, ensuring the kill switch is operating when appropriate, and that the transitions between states of the VPN are handled correctly.

A significant issue in this core logic could mean that traffic would not be properly routed through the VPN tunnel when it should be. Atredis Partners assessed this logic through consideration of the state machine, verifying that state transitions were complete, and ensuring that the system would only transition to insecure states at appropriate times. Any logic that could lead to the disabling of the "kill switch" rules that block insecure traffic was audited.

Dynamic testing was used to probe for traffic leaks in various edge cases. Tooling on the client operating systems, instrumentation in the clients themselves, and traffic analysis at the network level were used to expose leaks during various system state transitions, and to verify normal operation.

The client applications on all platforms exposed a very minimal attack surface for any external attacker seeking to influence the user's system. Any inter-process communication used was not exposed to the external network or local browsers. The communication with the Mullvad API followed secure principles, using pinned public certificates to verify a secure channel. It was noted that certificate revocation lists were not used for the pinned certificates, but application updates could be used to push new certificates to users in the event of a compromised certificate.

Desktop Clients

The Mullvad VPN clients for desktop systems contain privileged pieces that run at a root or administrator privilege level. This includes the `mullvad-daemon`, which needs this privilege to adjust firewall rules, create routes, and create the VPN tunnel.

The mechanisms used by the daemon to apply firewall rules and adjust routes were audited on each platform to test for completeness, ensuring functionality of the kill switch and proper blocking of insecure traffic leaks.

If a significant issue existed in the daemon, a local attacker could possibly use this issue to escalate their privilege level up to that of the daemon's. The areas where the daemon interacts with the system were audited, looking for any undue influence an unprivileged user could have on the daemon or the libraries it depends on. The daemon code was statically searched to find and note the local attack surface and trace influence. The handling of commands via inter-process communication was audited to ensure proper checks exist to verify the commands. Secure permissions on pipes, files, or loaded binaries were also verified dynamically.

The Mullvad VPN client for Windows depends on a kernel-space driver created by Mullvad to support the split-tunnel functionality. An audit of the driver showed that the device was created in a way so that only a privileged user could interact with the driver. This secured device means that even if issues exist in the driver, they would not be useful for gaining administrator privilege. Sufficient issues in the driver, however, could be used to inject unsigned and unchecked code into the kernel, which could bypass kernel integrity, insert a rootkit, or more. Because of this, the split-tunnel driver was also audited, looking for issues that could be triggered through interacting with the created device.

Mobile Clients

The VPN clients for the mobile platforms differ from the desktop platforms in that they are limited to using the mobile operating system's API to manage the VPN and have no fully privileged components. This eliminates the risk of a Mullvad component being used by a malicious party to escalate privilege, but also limits the protections Mullvad can provide to those actions the operating system's API supports.

At the time of the assessment, Mullvad was already aware of some limitations of the APIs and had documented the differences between the mobile platforms and desktop platforms in their security documentation.

When assessing the mobile clients, a dynamic testing environment was created to detect leaks while running the clients through various configurations and states. This environment consisted of passive traffic analysis at the tested device's network gateway, an internet host operated by Atredis, and both tooling and instrumentation at the client level, leveraging operating system access where feasible. Usage of the operating system-provided VPN APIs was also audited, ensuring proper configuration.

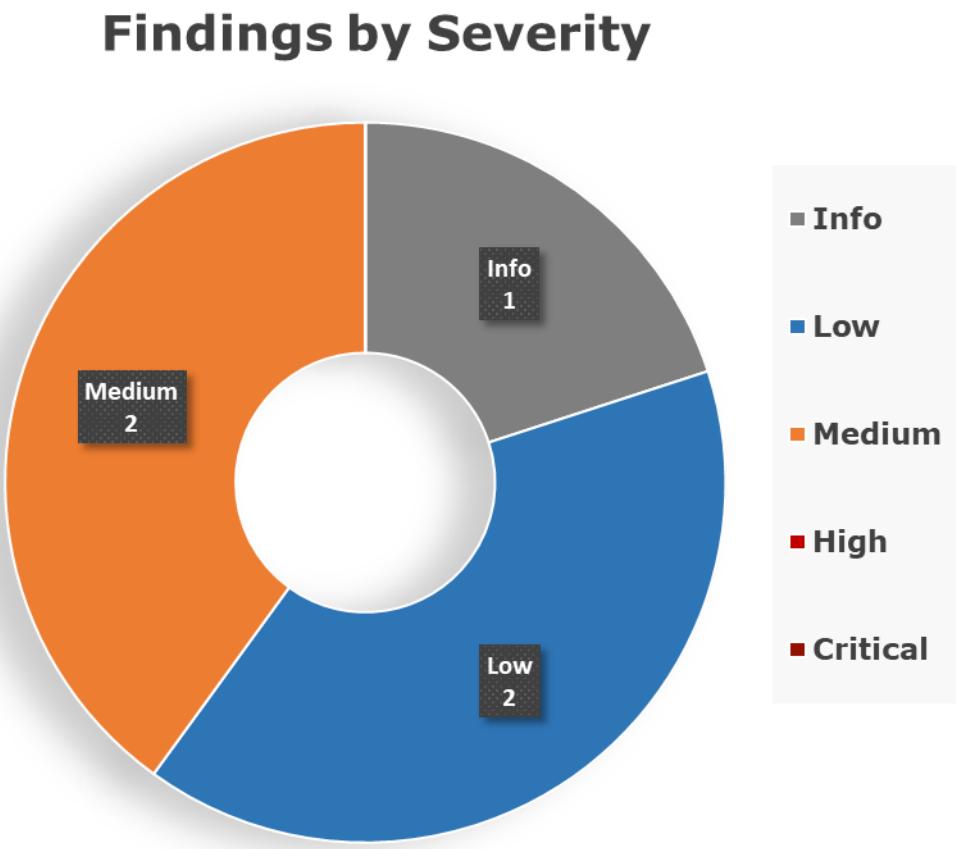
The issues discovered in the mobile clients were not solvable by Mullvad alone, but the Mullvad team updated documentation and created issues in the operating system vendor's issue trackers where appropriate.

Findings Summary

In performing testing for this assessment, Atredis Partners identified **two (2) medium**, **two (2) low** severity findings, and **one (1) informational** finding. No high or critical severity findings were noted. None of these issues constitute a potential for direct compromise or targeted leaking of arbitrary user traffic by an attacker.

Atredis defines vulnerability severity ranking as follows:

- **Critical:** These vulnerabilities expose systems and applications to immediate threat of compromise by a dedicated or opportunistic attacker.
- **High:** These vulnerabilities entail greater effort for attackers to exploit and may result in successful network compromise within a relatively short time.
- **Medium:** These vulnerabilities may not lead to network compromise but could be leveraged by attackers to attack other systems or applications components or be chained together with multiple medium findings to constitute a successful compromise.
- **Low:** These vulnerabilities are largely concerned with improper disclosure of information and should be resolved. They may provide attackers with important information that could lead to additional attack vectors or lower the level of effort necessary to exploit a system.



Remediation Tasks

The Mullvad team was prompt in responding to the issues discovered as they arose, and in each issue, we have noted the remediation actions the Mullvad team has taken. Where changes were made to the application, the changes were reviewed and tested to be complete. Some issues on the mobile platforms were due to limitations of the VPN API on those platforms, and as such the only current remediation involves updating documentation and working with the operating system vendors.

Findings and Recommendations

The following section outlines findings identified via manual and automated testing over the course of this engagement. Where necessary, specific artifacts to validate or replicate issues are included, as well as Atredis Partners' views on finding severity and recommended remediation.

Findings Summary

The below tables summarize the number and severity of the unique issues identified throughout the engagement.

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	0	2	2	1

Findings Detail

FINDING NAME	SEVERITY
MUL22-01: Out-of-Bounds Read in win-split-tunnel (Windows)	Low
MUL22-02: Leak of Traffic During System Shutdown	Medium
MUL22-03: Connectivity Checks Bypass VPN (Android)	Medium
MUL22-04: Permissive Inbound Network Filtering (Android)	Low
MUL22-05: Siri Shortcuts Susceptible to Manipulation (iOS)	Info

MUL22-01: Out-of-Bounds Read in win-split-tunnel (Windows)

Severity: Low

Finding Overview

The `mullvad-split-tunnel` driver contains an integer overflow leading to an out-of-bounds (OOB) read that could lead to memory corruption if an invalid device control request is sent to the driver. The vulnerability is mitigated by the access controls on the device, only allowing administrators to directly communicate with the driver.

Finding Detail

On the Windows platform, Mullvad provides split-tunnel support using a custom WDF (Windows Driver Frameworks) driver. This driver creates a device that handles device I/O control requests.

Requests with the control code `IOCTL_ST_SET_CONFIGURATION` take in a set of configuration entries and apply them. Before the incoming data is processed, the data is validated with a call to `ValidateUserBufferConfiguration`.

```
    /* ... */

    /* header is the beginning of the user controlled data
    auto header = (ST_CONFIGURATION_HEADER*)Buffer;

    if (header->TotalLength != BufferLength)
    {
        return false;
    }

    /* the stringBuffer should reside after the header and entries */
    /* the NumEntries is user controlled, and can be any SIZE_T number */
    /* An integer overflow can result in stringBuffer still pointing in the small buffer */
    /* Even if NumEntries is much larger than the buffer */
    auto stringBuffer = (UCHAR*)Buffer
        + sizeof(ST_CONFIGURATION_HEADER)
        + (sizeof(ST_CONFIGURATION_ENTRY) * header->NumEntries);

    if (stringBuffer < (UCHAR*)Buffer
        || stringBuffer >= bufferEnd)
    {
        return false;
    }

    /* ... */
    auto entry = (ST_CONFIGURATION_ENTRY*)(header + 1);

    for (auto i = 0; i < header->NumEntries; ++i, ++entry)
    {
        const auto valid = util::ValidateBufferRange(stringBuffer, bufferEnd,
            entry->ImageNameOffset, entry->ImageNameLength);

        if (!valid)
        {
            return false; /* Early return if invalid entry */
        }
    }

    /* ... */
```

A portion of `ValidateUserBufferConfiguration`, annotated

As annotated above, an integer overflow could exist while verifying the buffer, if the `NumEntries` value was, for example, `0x8000000000000000`. This would allow the looping and verifying of entries past the end of the buffer. The entry pointer will keep incrementing and using the undefined space past the end as `ST_CONFIGURATION_ENTRY` structures.

This vulnerability is not a high-priority vulnerability, as an attacker would have to be an administrator to reach the device exposed by `mullvad-split-tunnel` and would have to stop the `mullvad-daemon` first as well. Even for an authenticated attacker, the information leaked by the buffer over-read is limited and unstable.

Recommendation(s)

To improve the stability of the system, the driver should use safe integer functions when processing input from the user, such as `RtlSizeTMult`.

Fix Analysis

After receipt of the above information, the Mullvad team implemented a fix in both `ValidateUserBufferConfiguration` and `ValidateUserBufferProcesses`. These functions now use safe integer functions to avoid integer overflow. The fixed logic appears correct when validating offsets used in the buffers of those two request types, avoiding any memory corruption.

References

Merged Fix in win-split-tunnel Repository:
https://github.com/mullvad/win-split-tunnel/pull/34/files

CWE-190: Integer Overflow or Wraparound:
https://cwe.mitre.org/data/definitions/190.html

Safe Integer Functions:
https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/ntintsafe-design-guide

MUL22-02: Leak of Traffic During System Shutdown

Severity: Medium

Finding Overview

When a system is shut down while Mullvad is running, traffic may end up leaking outside the tunnel before the system shutdown completes. Systems may experience unexpected shutdowns due to battery life, overheating, or other events. These unexpected shutdowns could expose sensitive traffic that was intended for the VPN tunnel.

Finding Detail

The `mullvad-daemon` on Unix systems listens for `SIGTERM` or `SIGINT`. Upon receipt of one of these signals, the `mullvad-daemon` will remove the route changes and firewall rules, even if the auto-connect setting is set.

The `SIGTERM` signal is sent by Linux to each running process to indicate that the system is shutting down, followed by an uncatchable `SIGKILL`. When testing with a program that ignores `SIGTERM`, data was consistently leaked that would have been blocked. This data was leaked even with the auto-connect Mullvad setting enabled. While it is uncommon for programs to ignore `SIGTERM`, many programs will handle the signal to properly clean up, extending the window in which data may be leaked. There may also exist a small window in which regular programs with default signal handling will be able to leak data, depending on the order in which these programs and the `mullvad-daemon` are sent the termination signal.

This window of opportunity for leaked traffic may come as a surprise to users who experience a shutdown due to battery life, overheating, or other events.

Recommendation(s)

Firewall rules should not be removed during system shutdown. Letting firewall rules remain until power-off will block unintended leaks.

Fix Analysis

After receipt of the above information, the Mullvad team implemented fixes to this and other possibly related system power state leaks. The system now attempts to differentiate between a user-initiated shutdown of the daemon and a shutdown due to a power event. In the places where the type of shutdown is not known by the system, a safe default is chosen of denying traffic outside the tunnel.

References

Merged Fix Targeting Windows:
https://github.com/mullvad/mullvadvpn-app/pull/3942

Merged Fix Targeting Linux:
https://github.com/mullvad/mullvadvpn-app/pull/3940

Merged Fix Targeting macOS:
https://github.com/mullvad/mullvadvpn-app/pull/3943

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor:
https://cwe.mitre.org/data/definitions/200.html

MUL22-03: Connectivity Checks Bypass VPN (Android)

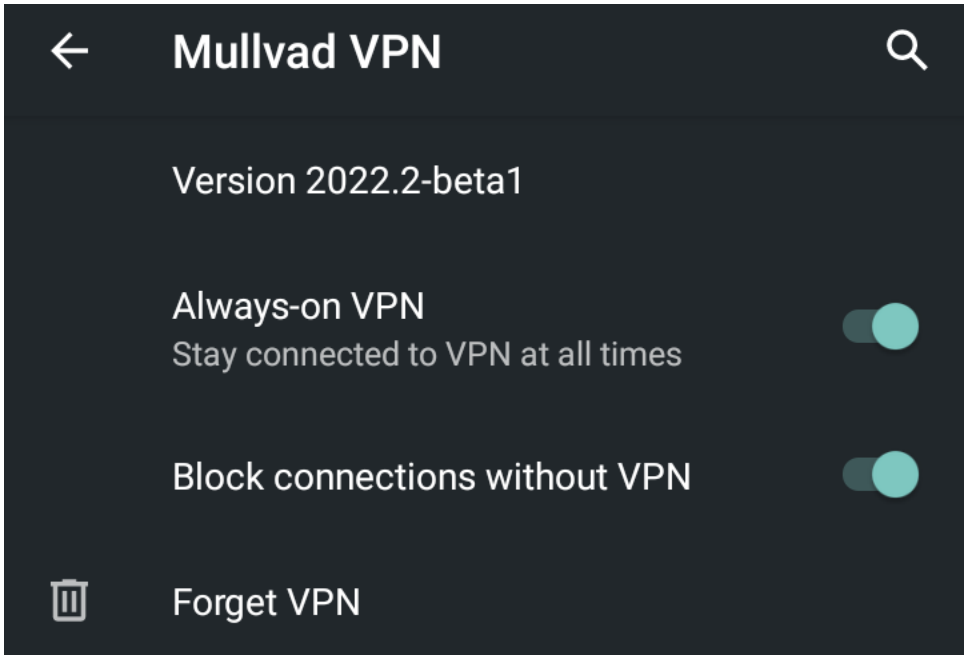
Severity: Medium

Finding Overview

Atredis discovered that the connectivity checks on Android bypass the configured VPN - regardless of whether non-VPN network traffic is blocked - leaking DNS (Domain Name System) and HTTP(S) (Hypertext Transfer Protocol (Secure)) requests to Google and potentially other entities. Connectivity checks are typically used for captive portal detection and to detect if the device has internet access.

Finding Detail

Atredis began testing this scenario by enabling "always-on" VPN and configuring it to block all non-VPN traffic.



"All" non-VPN traffic is blocked

Atredis then configured the `tcpdump` tool, a network "sniffer," to listen for any outbound traffic from the device (`192.168.1.9`). In the example below, `tcpdump` is running on the device's network gateway (`192.168.1.1`). Next, Atredis rebooted the device and observed non-VPN traffic coming from the device. (In testing, Mullvad and Atredis noted that reconnecting to the network also sent connectivity checks outside the VPN tunnel.)

```
# tcpdump -ni any "src host 192.168.1.9"
21:23:33.065215 IP 192.168.1.9.14578 > 192.168.1.1.53: 30754+ A? www.baidu.com. (31)
21:23:33.078157 IP 192.168.1.9.27544 > 192.168.1.1.53: 35666+ A? connectivitycheck.gstatic.
com. (47)
21:23:33.186947 IP 192.168.1.9.39376 > 183.232.231.173.443: Flags [S], …
21:23:33.201511 IP 192.168.1.9.43022 > 172.217.1.227.443: Flags [S], …
21:23:33.221978 IP 192.168.1.9.43022 > 172.217.1.227.443: Flags [.], …
21:23:33.230959 IP 192.168.1.9.43022 > 172.217.1.227.443: Flags [P.], …
21:23:33.234104 IP 192.168.1.9.46260 > 172.217.1.227.80: Flags [S], …
21:23:33.267781 IP 192.168.1.9.46260 > 172.217.1.227.80: Flags [.], …
21:23:33.388640 IP 192.168.1.9.43022 > 172.217.1.227.443: Flags [F.], …
21:23:33.409949 IP 192.168.1.9.39376 > 183.232.231.173.443: Flags [.], …
21:23:33.413449 IP 192.168.1.9.39376 > 183.232.231.173.443: Flags [P.], …
21:23:33.514252 IP 192.168.1.9.46260 > 172.217.1.227.80: Flags [P.], …
21:23:33.545633 IP 192.168.1.9.46260 > 172.217.1.227.80: Flags [F.], …
```

Outbound packets where non-VPN traffic should be blocked

As evidenced in the packet capture, DNS and HTTP(S) requests for `connectivitycheck.gstatic.com` (Google) and `www.baidu.com` (Baidu) effectively bypass the VPN when such traffic should be tunneled correctly or blocked outright.

Recommendation(s)

Since Android connectivity checks are tightly integrated with the operating system, there is little Mullvad's VPN software can do to change this core behavior. The official Android developer documentation suggests that *all* non-VPN traffic can be blocked, but this is clearly not the case in practice, where captive portal detection may otherwise be broken.

Fix Analysis

This issue is presently outside Mullvad's control, so there are no fixes to analyze currently. However, for the sake of transparency, Mullvad has reported the issue to Google and is working on a blog post to their users. Mullvad has already updated their `security.md` documentation to reflect this finding.

References

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor:
https://cwe.mitre.org/data/definitions/200.html

Android: Always-on VPN: Blocked connections:
https://developer.android.com/guide/topics/connectivity/vpn#blocked_connections

GitHub: Document Android leaks reported in MUL22-03:
https://github.com/mullvad/mullvadvpn-app/commit/c078c907b6f14959b1924741b0e5781c20562f9f

Issues added to the Android issue tracker by Mullvad:
https://issuetracker.google.com/issues/249990229
https://issuetracker.google.com/issues/250529027

A blog post by Mullvad on this leak:
https://mullvad.net/en/blog/2022/10/10/android-leaks-connectivity-check-traffic/

MUL22-04: Permissive Inbound Network Filtering (Android)

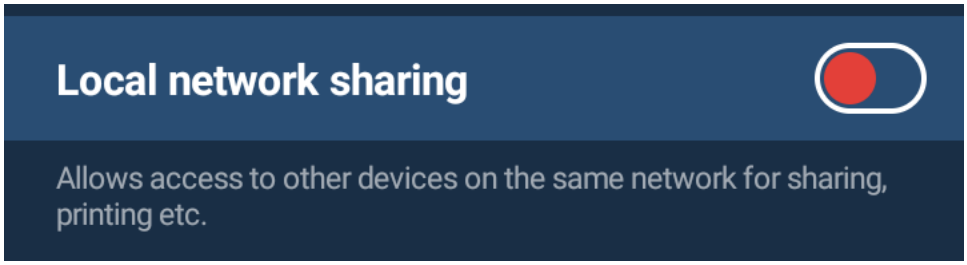
Severity: Low

Finding Overview

When local network sharing is disabled on Android, inbound traffic from the local network is still permitted, including new connections to the device. While not considered a leak, this behavior is inconsistent with the behavior on other platforms where the firewall is utilized to block such traffic.

Finding Detail

Atredis began testing this scenario by enabling the VPN with local network sharing disabled. It was verified that traffic to the internet was tunneled correctly, and traffic to the local network was blocked.



Local network sharing is disabled

Pinging and port scanning the device on the local network showed that the device was responding to all inbound traffic. An Android Debug Bridge (ADB) shell was used to perform further network testing.

```
wvu@hiigara:~$ ping 192.168.1.9
PING 192.168.1.9 (192.168.1.9): 56 data bytes
64 bytes from 192.168.1.9: icmp_seq=0 ttl=64 time=5.346 ms
64 bytes from 192.168.1.9: icmp_seq=1 ttl=64 time=6.278 ms
64 bytes from 192.168.1.9: icmp_seq=2 ttl=64 time=100.804 ms
64 bytes from 192.168.1.9: icmp_seq=3 ttl=64 time=5.394 ms
^C
--- 192.168.1.9 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 5.346/29.456/100.804/41.195 ms
wvu@hiigara:~$
```

Pinging the device on the local network

Using a `netcat` listener on the device, Atredis confirmed that new connections to the device through the local network were still allowed, while outbound traffic to the local network was still being blocked.

```
wvu@hiigara:~$ ncat -v 192.168.1.9 4444
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Connected to 192.168.1.9:4444.
Atredis was here.
```

Connecting to the device on the local network

```
wvu@hiigara:~$ adb shell
S42:/ $ nc -lp 4444
Atredis was here.
```

Receiving the connection on the device

```
S42:/ $ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3092ms

1|S42:/ $
```

Pinging a local network host from the device

Recommendation(s)

The behavior of the local network sharing feature should be consistent across supported platforms. When local network sharing is disabled, local network traffic to and from the device should be blocked. Atredis understands that the current behavior may be a limitation of relying on the system's VPN service API to perform network filtering. This caveat is mentioned in the official documentation (referenced below).

Fix Analysis

This issue is presently outside Mullvad's control, so there are no fixes to analyze at this time. However, for the sake of transparency, Mullvad has updated their `security.md` documentation to reflect this finding.

References

CWE-183: Permissive List of Allowed Inputs:
https://cwe.mitre.org/data/definitions/183.html

GitHub: Mullvad VPN app security: Desktop vs mobile:
https://github.com/mullvad/mullvadvpn-app/blob/master/docs/security.md#desktop-vs-mobile

GitHub: Document Android shortcoming:
https://github.com/mullvad/mullvadvpn-app/commit/652e20aadb802fe03585b4f61fe501d03477b046

MUL22-05: Siri Shortcuts Susceptible to Manipulation (iOS)

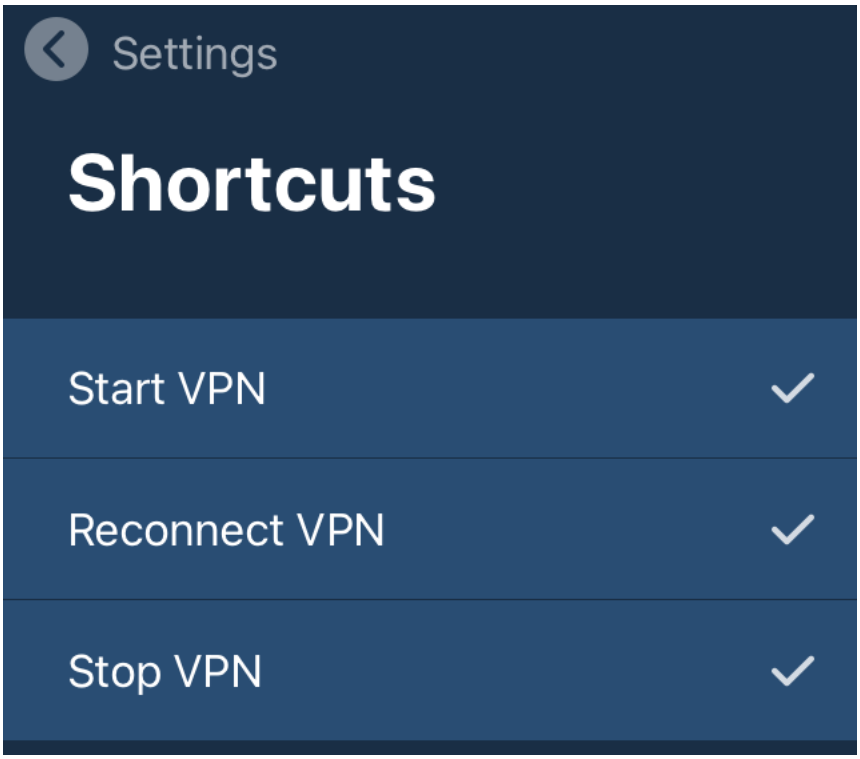
Severity: Info

Finding Overview

The Siri Shortcuts feature on iOS is susceptible to manipulation by an external attacker. An attacker with knowledge of Mullvad's voice commands may be able to control the VPN through multiple means, including but not limited to playing audio files or leveraging speech synthesis APIs. Note that the Shortcuts feature is strictly opt-in and not enabled by default in Mullvad's UI.

Finding Detail

Atredis developed the following proof of concept (PoC) to demonstrate disconnecting the VPN from within the user's web browser (Safari). Prerequisites include enabling Siri and enabling the Mullvad-specific Shortcuts inside Mullvad's settings. The device's volume should be turned up, too.



Mullvad-specific Shortcuts are enabled

Atredis began creating the PoC by using macOS' `say(1)` command to synthesize the voice command *"hey siri stop vpn"* and write the audio to the `poc.aiff` file. Atredis then used FFmpeg to convert the AIFF file to WAV format (for portability and testing), resulting in the `poc.wav` file used in subsequent steps.

```
wvu@hiigara:~/Downloads/poc$ echo "hey siri stop vpn" | say -o poc.aiff
wvu@hiigara:~/Downloads/poc$ file poc.aiff
poc.aiff: IFF data, AIFF-C compressed audio
wvu@hiigara:~/Downloads/poc$ ffmpeg -v quiet -i poc.aiff poc.wav
wvu@hiigara:~/Downloads/poc$ file poc.wav
poc.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 22050 Hz
wvu@hiigara:~/Downloads/poc$
```

Saying "hey siri stop vpn" using a computer

Atredis wrote the following HTML (Hypertext Markup Language) into a new `index.html` file, sourcing the `poc.wav` file generated earlier. The code displays a play button for the `poc.wav` file and attempts to autoplay the file, though this typically fails in modern browsers due to autoplay restrictions.

```
<audio controls autoplay>
  <source src="poc.wav" type="audio/wav">
</audio>
```

The contents of the `index.html` file

Finally, Atredis spawned a portable web server using Python and served the previously created files at URL `http://192.168.1.6/` (for demonstration purposes), which the user (`192.168.1.17`) then visited in the Safari browser. Upon clicking the play button on the page, the `poc.wav` audio file was played, the Siri agent on the device responded to the audio, and the user was promptly disconnected from VPN.

```
wvu@hiigara:~/Downloads/poc$ python3 -m http.server 80
Serving HTTP on :: port 80 (http://[::]:80/) ...
::ffff:192.168.1.17 - - [04/Oct/2022 16:29:17] "GET / HTTP/1.1" 200 -
::ffff:192.168.1.17 - - [04/Oct/2022 16:29:17] "GET /poc.wav HTTP/1.1" 200 -
```

Spawning a Python web server to deliver the PoC

Recommendation(s)

Since this issue is the misuse of a system feature, few recommendations can be given to Mullvad or their users. The feature is already opt-in, meaning it is disabled by default and requires explicit permission to enable. For users, the only realistic recommendation is to change the default voice commands to something less predictable - or simply leave the feature (or specific shortcuts) disabled.

Fix Analysis

This issue is presently outside Mullvad's control, so there are no fixes to analyze at this time.

References

CWE-1039: Automated Recognition Mechanism with Inadequate Detection or Handling of Adversarial Input Perturbations:
https://cwe.mitre.org/data/definitions/1039.html

Apple: Run shortcuts with Siri, the Shortcuts app, or Siri Suggestions:
https://support.apple.com/en-us/HT209055

GitHub: Adding support for Siri Shortcuts in iOS:
https://github.com/mullvad/mullvadvpn-app/issues/2897

Appendix I: Assessment Methodology

Atredis Partners draws on our extensive experience in penetration testing, reverse engineering, hardware/software exploitation, and embedded systems design to tailor each assessment to the specific targets, attacker profile, and threat scenarios relevant to our client's business drivers and agreed upon rules of engagement.

Where applicable, we also draw on and reference specific industry best practices, regulations, and principles of sound systems and software design to help our clients improve their products while simultaneously making them more stable and secure.

Our team takes guidance from industry-wide standards and practices such as the National Institute of Standards and Technology's (NIST) Special Publications, the Open Web Application Security Project (OWASP), and the Center for Internet Security (CIS).

Throughout the engagement, we communicate findings as they are identified and validated, and schedule ongoing engagement meetings and touchpoints, keeping our process open and transparent and working closely with our clients to focus testing efforts where they provide the most value.

In most engagements, our primary focus is on creating purpose-built test suites and toolchains to evaluate the target, but we do utilize off-the-shelf tools where applicable as well, both for general patch audit and best practice validation as well as to ensure a comprehensive and consistent baseline is obtained.

Research and Profiling Phase

Our research-driven approach to testing begins with a detailed examination of the target, where we model the behavior of the application, network, and software components in their default state. We map out hosts and network services, patch levels, and application versions. We frequently use a number of private and public data sources to collect Open Source Intelligence about the target, and collaborate with client personnel to further inform our testing objectives.

For network and web application assessments, we perform network and host discovery as well as map out all available application interfaces and inputs. For hardware assessments, we study the design and implementation, down to a circuit-debugging level. In reviewing source code or compiled application code, we map out application flow and call trees and develop a solid working understand of how the application behaves, thus helping focus our validation and testing efforts on areas where vulnerabilities might have the highest impact to the application's security or integrity.

Analysis and Instrumentation Phase

Once we have developed a thorough understanding of the target, we use a number of specialized and custom-developed tools to perform vulnerability discovery as well as binary, protocol, and runtime analysis, frequently creating engagement-specific software tools which we share with our clients at the close of any engagement.

We identify and implement means to monitor and instrument the behavior of the target, utilizing debugging, decompilation and runtime analysis, as well as making use of memory and filesystem

forensics analysis to create a comprehensive attack modeling testbed. Where they exist, we also use common off-the-shelf, open-source and any extant vendor-proprietary tools to aid in testing and evaluation.

Validation and Attack Phase

Using our understanding of the target, our team creates a series of highly-specific attack and fault injection test cases and scenarios. Our selection of test cases and testing viewpoints are based on our understanding of which approaches are most relevant to the target and will gain results in the most efficient manner, and built in collaboration with our client during the engagement.

Once our test cases are validated and specific attacks are confirmed, we create proof-of-concept artifacts and pursue confirmed attacks to identify extent of potential damage, risk to the environment, and reliability of each attack scenario. We also gather all the necessary data to confirm vulnerabilities identified and work to identify and document specific root causes and all relevant instances in software, hardware, or firmware where a given issue exists.

Education and Evidentiary Phase

At the conclusion of active testing, our team gathers all raw data, relevant custom toolchains, and applicable testing artifacts, parses and normalizes these results, and presents an initial findings brief to our clients, so that remediation can begin while a more formal document is created. Additionally, our team shares confirmed high-risk findings throughout the engagement so that our clients may begin to address any critical issues as soon as they are identified.

After the outbrief and initial findings review, we develop a detailed research deliverable report that provides not only our findings and recommendations but also an open and transparent narrative about our testing process, observations and specific challenges in developing attacks against our targets, from the real world perspective of a skilled, motivated attacker.

Automation and Off-The-Shelf Tools

Where applicable or useful, our team does utilize licensed and open-source software to aid us throughout the evaluation process. These tools and their output are considered secondary to manual human analysis, but nonetheless provide a valuable secondary source of data, after careful validation and reduction of false positives.

For runtime analysis and debugging, we rely extensively on Hopper, IDA Pro and Hex-Rays, as well as platform-specific runtime debuggers, and develop fuzzing, memory analysis, and other testing tools primarily in Ruby and Python.

In source auditing, we typically work in Visual Studio, Xcode and Eclipse IDE, as well as other markup tools. For automated source code analysis we will typically use the most appropriate toolchain for the target, unless client preference dictates another tool.

Network discovery and exploitation make use of Nessus, Metasploit, and other open-source scanning tools, again deferring to client preference where applicable. Web application runtime analysis relies extensively on the Burp Suite, Fuzzer and Scanner, as well as purpose-built automation tools built in Go, Ruby and Python.

Engagement Deliverables

Atredis Partners deliverables include a detailed overview of testing steps and testing dates, as well as our understanding of the specific risk profile developed from performing the objectives of the given engagement.

In the engagement summary we focus on "big picture" recommendations and a high-level overview of shared attributes of vulnerabilities identified and organizational-level recommendations that might address these findings.

In the findings section of the document, we provide detailed information about vulnerabilities identified, provide relevant steps and proof-of-concept code to replicate these findings, and our recommended approach to remediate the issues, developing these recommendations collaboratively with our clients before finalization of the document.

Our team typically makes use of both DREAD and NIST CVE for risk scoring and naming, but as part of our charter as a client-driven and collaborative consultancy, we can vary our scoring model to a given client's preferred risk model, and in many cases will create our findings using the client's internal findings templates, if requested.

Sample deliverables can be provided upon request, but due to the highly specific and confidential nature of Atredis Partners' work, these deliverables will be heavily sanitized, and give only a very general sense of the document structure.

Appendix II: Engagement Team Biographies

Shawn Moyer, Founding Partner and CEO

Shawn Moyer scopes, plans, and coordinates security research and consulting projects for the Atredis Partners team, including reverse engineering, binary analysis, advanced penetration testing, and private vulnerability research. As CEO, Shawn works with the Atredis leadership team to build and grow the Atredis culture, making Atredis Partners a home for some of the best minds in information security, and ensuring Atredis continues to deliver research and consulting services that exceed our client's expectations.

Experience

Shawn brings over 25 years of experience in information security, with an extensive background in penetration testing, advanced security research including extensive work in mobile and Smart Grid security, as well as advanced threat modeling and embedded reverse engineering.

Shawn has served as a team lead and consultant in enterprise security for numerous large initiatives in the financial sector and the federal government, including IBM Internet Security Systems' X-Force, MasterCard, a large Federal agency, and Wells Fargo Securities, all focusing on emerging network and application attacks and defenses.

In 2010, Shawn created Accuvant Labs' Applied Research practice, delivering advanced research-driven consulting to numerous clients on mobile platforms, critical infrastructure, medical devices and countless other targets, growing the practice 1800% in its first year.

Prior to Accuvant, Shawn helped develop FishNet Security's penetration testing team as a principal security consultant, growing red team offerings and advanced penetration testing services, while being twice selected as a consulting MVP.

Key Accomplishments

Shawn has written on emerging threats and other topics for Information Security Magazine and ZDNet, and his research has been featured in the Washington Post, BusinessWeek, NPR and the New York Times. Shawn is a twelve-time speaker at the Black Hat Briefings and has been an invited speaker at other notable security conferences around the world.

Shawn is likely best known for delivering the first public research on social network security, pointing out much of the threat landscape still exists on social network platforms today. Shawn also co-authored an analysis of the state of the art in web browser exploit mitigation, creating the first in-depth comparison of browser security models along with Dr. Charlie Miller, Chris Valasek, Ryan Smith, Joshua Drake, and Paul Mehta.

Shawn studied Computer and Network Information Systems at Missouri University and the University of Louisiana at Lafayette, holds numerous information security certifications, and has been a frequent presenter at national and international security industry conferences.

Shawn Moyer, Founding Partner and CEO

Shawn Moyer scopes, plans, and coordinates security research and consulting projects for the Atredis Partners team, including reverse engineering, binary analysis, advanced penetration testing, and private vulnerability research. As CEO, Shawn works with the Atredis leadership team to build and grow the Atredis culture, making Atredis Partners a home for some of the best minds in information security, and ensuring Atredis continues to deliver research and consulting services that exceed our client's expectations.

Experience

Shawn brings over 25 years of experience in information security, with an extensive background in penetration testing, advanced security research including extensive work in mobile and Smart Grid security, as well as advanced threat modeling and embedded reverse engineering.

Shawn has served as a team lead and consultant in enterprise security for numerous large initiatives in the financial sector and the federal government, including IBM Internet Security Systems' X-Force, MasterCard, a large Federal agency, and Wells Fargo Securities, all focusing on emerging network and application attacks and defenses.

In 2010, Shawn created Accuvant Labs' Applied Research practice, delivering advanced research-driven consulting to numerous clients on mobile platforms, critical infrastructure, medical devices and countless other targets, growing the practice 1800% in its first year.

Prior to Accuvant, Shawn helped develop FishNet Security's penetration testing team as a principal security consultant, growing red team offerings and advanced penetration testing services, while being twice selected as a consulting MVP.

Key Accomplishments

Shawn has written on emerging threats and other topics for Information Security Magazine and ZDNet, and his research has been featured in the Washington Post, BusinessWeek, NPR and the New York Times. Shawn is a twelve-time speaker at the Black Hat Briefings and has been an invited speaker at other notable security conferences around the world.

Shawn is likely best known for delivering the first public research on social network security, pointing out much of the threat landscape still exists on social network platforms today. Shawn also co-authored an analysis of the state of the art in web browser exploit mitigation, creating the first in-depth comparison of browser security models along with Dr. Charlie Miller, Chris Valasek, Ryan Smith, Joshua Drake, and Paul Mehta.

Shawn studied Computer and Network Information Systems at Missouri University and the University of Louisiana at Lafayette, holds numerous information security certifications, and has been a frequent presenter at national and international security industry conferences.

Jordan Whitehead, Principal Research Consultant

Jordan Whitehead specializes in vulnerability research and binary exploitation. Jordan is able to quickly dive into large systems and find key weaknesses as a result of his significant experience in operating system internals.

Experience

During Jordan's Computer Engineering degree schooling, he created and instructed collegiate courses and clubs on computer security. After college he worked as a CNO developer for ManTech International, developing tools and capabilities that involved deep exploration into modern operating systems for exploitable weaknesses. While in that position, Jordan also continued to help create and instruct a number of formal reverse engineering and exploitation courses. These courses detailed the system internals for Windows, Linux, and Android. He has worked with research teams developing custom virtualization and emulation tooling that enabled researchers to better assess otherwise unreachable systems.

Key Accomplishments

Jordan has helped publish papers at top academic conferences on computer security, including Usenix Security Symposium. He has also developed open-source tools related to vulnerability research and secure software. These include peer-reviewed tools that have helped provide useable security and trust on Linux and Windows platforms.

William Vu, Senior Research Consultant

William Vu specializes in vulnerability research and exploit development of software systems and networks.

Experience

William has nearly a decade of professional experience in the information security industry. Prior to joining Atredis Partners, William was a founding member of Rapid7's vulnerability research team and a core developer for the Metasploit Project.

William's experience includes system administration, software engineering, red teaming, and digital forensics.

Key Accomplishments

William has published highly technical security research, numerous exploits, and discovered multiple vulnerabilities in well-known products. In addition to published content, William has presented at the DEF CON security conference and assisted trainings at Black Hat USA.

Nicholas Nam, Research Consulting Director

Nicholas Nam leads and executes highly technical network and application security assessments, as well as adversarial simulation assessments and other advanced research consulting engagements.

Experience

Nick has 19 years of professional experience in the information security space, where the early part of his career was focused on defensive roles at various organizations within the U.S. Department of Defense. Nick has spent the last 8 years within the private sector performing offensive security testing against web applications and networks as well as adversary simulations.

Prior to joining Atredis Partners, Nick was a Security Innovation Principal with FusionX (an Accenture company). While at FusionX, Nick performed many goal-based adversary simulations against some the world's top communications, financial and media companies.

Key Accomplishments

As part of his work performing adversary simulations, Nick has successfully taken over a national television network's broadcast system, payroll and wire transfer systems for several large, multinational financial institutions, and gained unrestricted access to secret projects such as proprietary genomic data at a large multinational chemical company.

Sara Bettes, Client Operations Associate

Sara Bettes assists the creation and completion of projects at Atredis Partners, ranging from the full pre-sales process to project design and management, to final delivery and follow-up. Her goals are to ensure all projects are executed in a way that reaches the goals of the client and assists the consultants at every turn.

Experience

Prior to joining Atredis Partners, Sara led a team that planned international sporting competitions, Olympic and national team qualifying events, as well as supported the mission of multiple non-profits. Her experience includes Live Sports Commentating, Staffing Management, Safety Plan Creation, Event Development, Public Relations, and Marketing efforts.

Key Accomplishments

Sara earned a bachelor's degree in Mass Communications with an emphasis in Broadcast and Public Relations from Oklahoma City University.

Appendix III: About Atredis Partners

Atredis Partners was created in 2013 by a team of security industry veterans who wanted to prioritize offering quality and client needs over the pressure to grow rapidly at the expense of delivery and execution. We wanted to build something better, for the long haul.

In six years, Atredis Partners has doubled in size annually, and has been named three times to the Saint Louis Business Journal's "Fifty Fastest Growing Companies" and "Ten Fastest Growing Tech Companies". Consecutively for the past three years, Atredis Partners has been listed on the Inc. 5,000 list of fastest growing private companies in the United States.

The Atredis team is made up of some of the greatest minds in Information Security research and penetration testing, and we've built our business on a reputation for delivering deeper, more advanced assessments than any other firm in our industry.

Atredis Partners team members have presented research over forty times at the BlackHat Briefings conference in Europe, Japan, and the United States, as well as many other notable security conferences, including RSA, ShmooCon, DerbyCon, BSides, and PacSec/CanSec. Most of our team hold one or more advanced degrees in Computer Science or engineering, as well as many other industry certifications and designations. Atredis team members have authored several books, including *The Android Hacker's Handbook*, *The iOS Hacker's Handbook*, *Wicked Cool Shell Scripts*, *Gray Hat C#*, and *Black Hat Go*.

While our client base is by definition confidential and we often operate under strict nondisclosure agreements, Atredis Partners has delivered notable public security research on improving the security at Google, Microsoft, The Linux Foundation, Motorola, Samsung and HTC products, and were the first security research firm to be named in Qualcomm's Product Security Hall of Fame. We've received four research grants from the Defense Advanced Research Project Agency (DARPA), participated in research for the CNCF (Cloud Native Computing Foundation) to advance the security of Kubernetes, worked with OSTIF (The Open Source Technology Improvement Fund) and The Linux Foundation on the Core Infrastructure Initiative to improve the security and safety of the Linux Kernel, and have identified entirely new classes of vulnerabilities in hardware, software, and the infrastructure of the World Wide Web.

In 2015, we expanded our services portfolio to include a wide range of advanced risk and security program management consulting, expanding our services reach to extend from the technical trenches into the boardroom. The Atredis Risk and Advisory team has extensive experience building mature security programs, performing risk and readiness assessments, and serving as trusted partners to our clients to ensure the right people are making informed decisions about risk and risk management.