tinexta defence

CVE-2024-4944 A Local Privilege escalation in WatchGuard Mobile VPN with SSL

#TinextaDefenceBusiness

Malware Lab

Summary

Our Malware Lab

Tinexta Defence Malware Lab daily performs dissection of malware with the aim of timely understanding the technological evolutions of attacks, consolidating the knowledge of necessary to make more effective and faster the process of incidents responding, contributing to spreading information about emerging threats into the expert's community and among its clients.

Malware Lab analysts are continuously engaged in searching and experimenting new analysis tools, for increasing accuracy and scope of action with regard to the proliferation of new evasion and anti-analysis techniques adopted by malware.

The Malware Lab is also committed to the development of proprietary tools for malware analysis and supporting the management and response of incidents.

Besides malware analysis, Malware Lab ideated and implemented an automatic process of extraction of **Indicators of Compromise (IOC)** that is daily run on dozens of new malwares, intercepted in the wide for populating our Knowledge Base.

Corrado Aaron Visaggio
Group Chief Scientist Officer & Malware Lab Director
a.visaggio@defencetech.it

1. Executive Summary

Our malware lab team discovered a local privilege escalation (LPE) vulnerability in WatchGuard Mobile VPN with SSL, a widely used VPN client for Windows.

This vulnerability allows attackers with basic user privileges to escalate to SYSTEM level on VPN-enabled endpoints, gaining unrestricted control over systems with direct corporate network access. The elevated privileges enable credential harvesting for lateral movement, bypass of security controls (DLP, EDR), and establishment of persistent backdoors that survive remediation attempts. Compromised VPN endpoints become critical pivot points for supply chain attacks potentially affecting partners and clients, while exposing organizations to regulatory violations and compliance breaches. The risk is amplified for remote worker devices operating outside the corporate security perimeter, where limited detection capabilities provide attackers with extended operational windows.

After responsibly disclosing the vulnerability to WatchGuard, they acknowledged the issue and released a patch in version 12.11.3 of their software. The vulnerability has been assigned CVE-2024-4944[1].

We encourage all users of the affected software to update to the latest version available in order to mitigate the risk associated with this vulnerability.

[1] https://www.watchguard.com/wgrd-psirt/advisory/wgsa-2025-00016

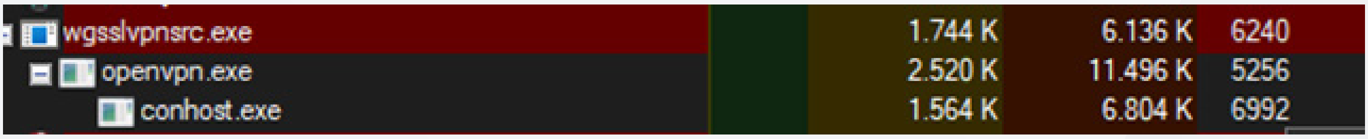
2. Vulnerability analysis

This research started while analyzing potential privilege escalation pathways in common applications used in business environments. We evaluated the list of common Windows software and focused on applications that need some kind of elevated privileges to perform their tasks.

We switched our focus to VPN software since it is extremely common on employee endpoints and most importantly creates new network interfaces, which usually requires administrative privileges. Since the user interface allows any regular user to connect/disconnect from the VPN, it means there must be some mechanism that allows the non-privileged control panel process to elevate its privileges and enable the virtual network interface.

For this research we worked on the WatchGuard Mobile VPN with SSL client since it is very popular in businesses and we had access to both the desktop application and, most importantly, a physical WatchGuard firewall for it to connect to.

In this case, discovering the elevation mechanism was very simple: using sysinternals Process Explorer[2] we could watch for the creation of new processes right after clicking the "Connect" button on the VPN client. This quickly leads us to discover a service called wgsslvpnsrc.exe spawning an instance of openvpn.exe as seen in the figure below.



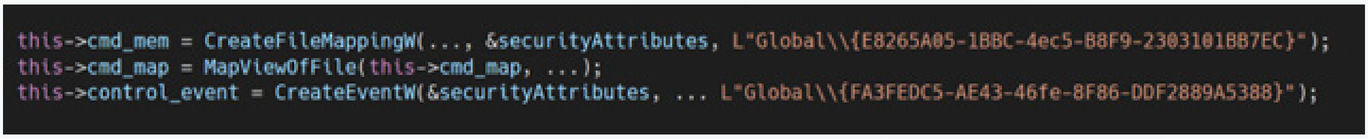
```
wgsslvpnsrc.exe          1.744 K     6.136 K   6240
  openvpn.exe            2.520 K    11.496 K   5256
    conhost.exe          1.564 K     6.804 K   6992
```

Both the service and *openvpn* run as Local System, meaning they have full administrative privileges on the endpoint. If we could find a way to abuse this service to run arbitrary commands, we would have a full privilege escalation primitive.

[2] https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer

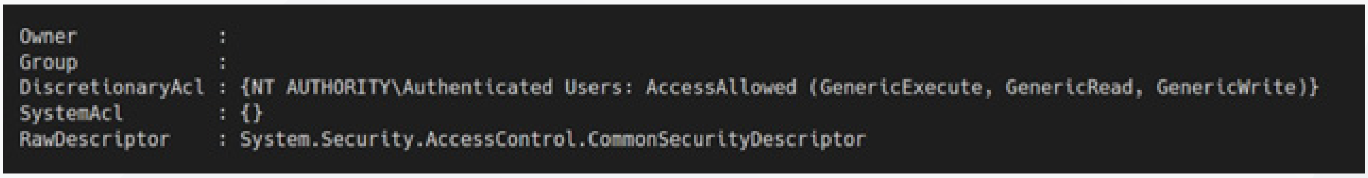
At this point we proceeded with reverse engineering the service binary; this revealed a peculiar communication mechanism between the VPN client and the service. The service creates several fixed-size shared memory sections identified by a hard-coded GUID; each contains a single string used for various connection parameters. Finally, a global named event is created to signal the service that new parameters are available in the shared memory sections.

The pseudo-code initialization looks like this:



```
this->cmd_mem = CreateFileMappingW(..., &securityAttributes, L"Global\\{E8265A05-1BBC-4ec5-B8F9-2303101BB7EC}");
this->cmd_map = MapViewOfFile(this->cmd_map, ...);
this->control_event = CreateEventW(&securityAttributes, ... L"Global\\{FA3FEDC5-AE43-46fe-8F86-DDF2889A5388}");
```

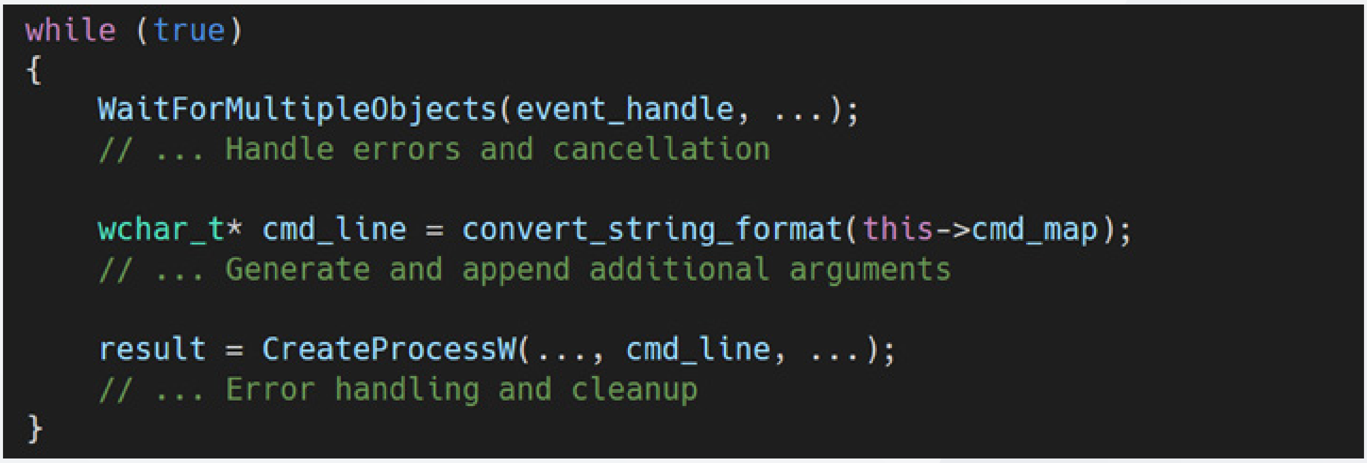
Where `securityAttributes` is initialized from the `D:(A;OICI;GRGWGX;;;AU)` SSDL string. We can convert this to a human readable format using the PowerShell utility `ConvertFrom-SddlString`:



```
Owner             :
Group             :
DiscretionaryAcl : {NT AUTHORITY\Authenticated Users: AccessAllowed (GenericExecute, GenericRead, GenericWrite)}
SystemAcl         : {}
RawDescriptor     : System.Security.AccessControl.CommonSecurityDescriptor
```

This means that any authenticated user on the system can open the shared memory sections and the event handle, thus allowing the desktop VPN client to communicate with the service.

Finally, the service launches an endless loop waiting for the event to be signaled. When this happens, it reads the various parameters and proceeds to directly execute the content of one of the shared memory sections using `CreateProcessW`.



```
while (true)
{
    WaitForMultipleObjects(event_handle, ...);
    // ... Handle errors and cancellation

    wchar_t* cmd_line = convert_string_format(this->cmd_map);
    // ... Generate and append additional arguments

    result = CreateProcessW(..., cmd_line, ...);
    // ... Error handling and cleanup
}
```

This allows a malicious client to write an arbitrary executable path into the shared memory section and signal the event, thus causing the service to execute the command as Local System. Exploitation is trivial and we provided WatchGuard with a complete PoC.

Detection of exploitation attempts is possible by monitoring the creation of processes spawned by `wgsslvpnsrc.exe` anything other than the openvpn.exe binary bundled with the VPN client should be considered suspicious.

After some back and forth with the vendor, the issue got assigned CVE-2024-4944 and patched in recent versions of WatchGuard Mobile VPN with SSL.

3. Conclusions

This vulnerability highlights the risks associated with insecure inter-process communication mechanisms. Developers should ensure that all shared resources are adequately protected, and that interfaces exposed to low privileged processes are properly hardened against potential abuse.

Users of WatchGuard Mobile VPN with SSL are strongly advised to update to the latest version available.

This kind of issue highlights the importance of regular security assessments and code reviews, particularly for software that operates with elevated privileges. Organizations should deploy monitoring solutions such as EDRs to detect anomalous behavior indicative of exploitation attempts as well as strong log retention policies to retroactively investigate potential exploitation of such vulnerabilities.

tinexta
defence

Next | Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 - 00131 Roma
tel. 06.45752720 - info@defencetech.it
www.tinextadefence.it

#TinextaDefenceBusiness