

Evading Antivirus Heuristics



William Hardy
Security Engineer



Module Overview



Evading API monitoring

Anti-emulation techniques

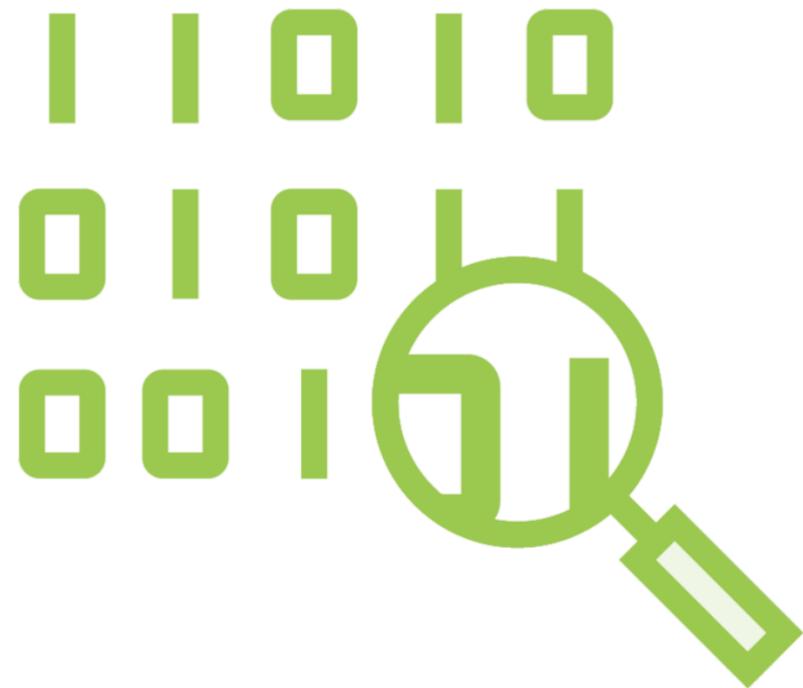
Anti-sandboxing techniques

Bypassing AMSI

Demo



Evading API Monitoring



API monitoring is performed in user-land by performing DLL injection and hooking APIs



The DLL is injected into our malicious process. We have control over our process! We can remove the hooks or avoid them



Evading Hooks



Patching the hook to restore the original function's entrypoint



Copying the code from the original DLL on disk and overwriting it in memory



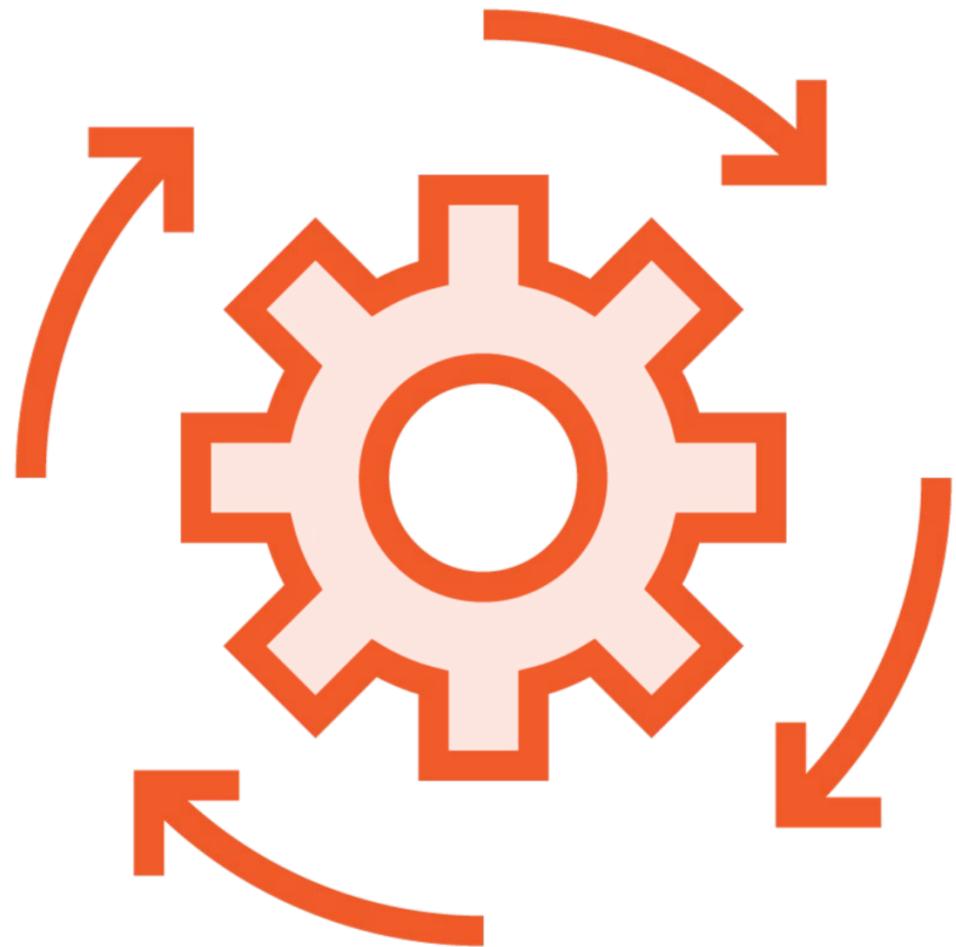
Copying the original DLL and reloading it at runtime



Performing manual syscalls



Anti-Emulation



Conditional code and un-emulated APIs

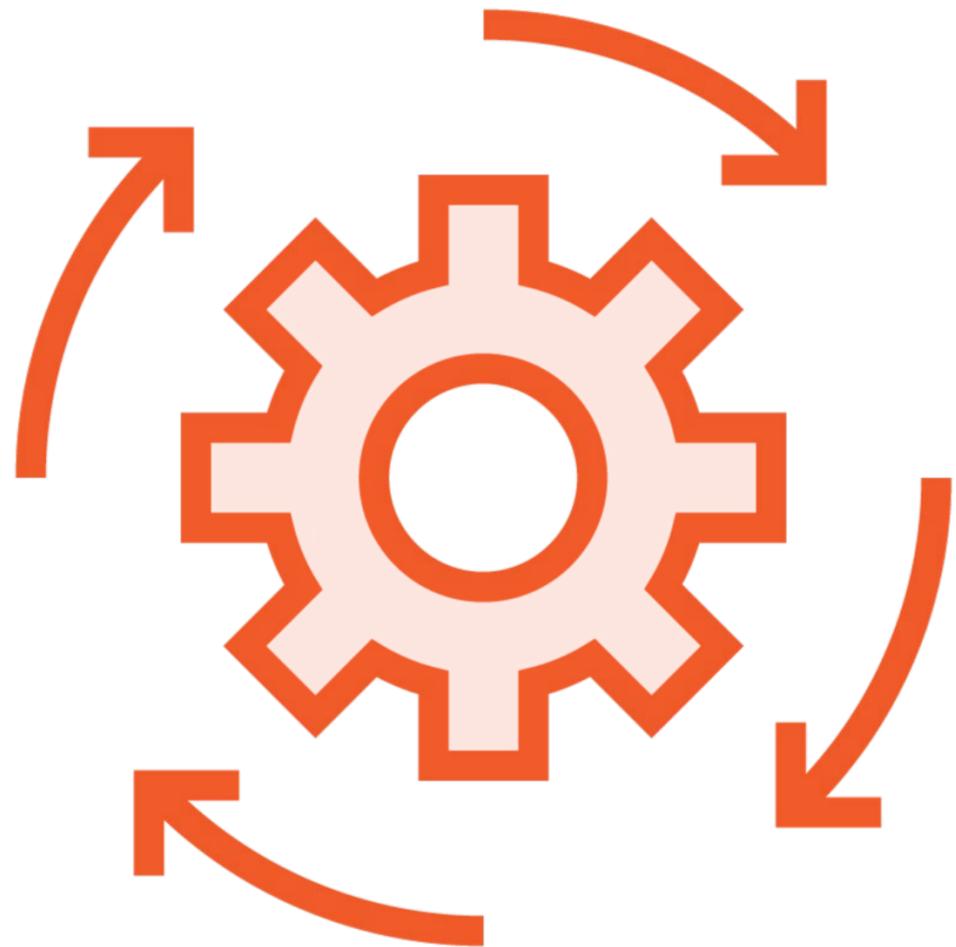
Exhausting the emulator

Conditional code which attempts to identify if emulation is occurring

Checking CPU registry value discrepancies



Anti-Sandboxing



Similar techniques as those covered in anti-emulation

Resource consumption is usually not a vector since it is probably in the Cloud

Fingerprinting the sandbox



Bypassing AMSI



Occurs inside your PowerShell process

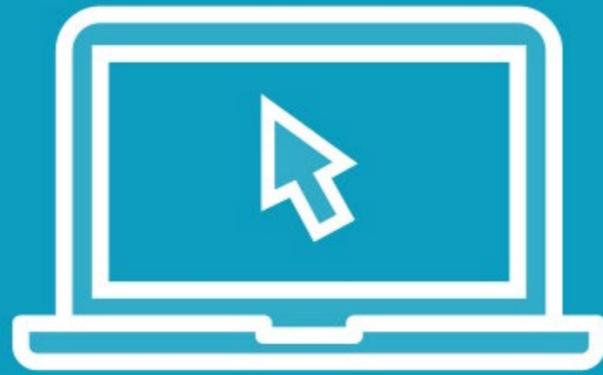
Automatically sends buffers containing code to be scanned to the AMSI API

The AMSI API can be tampered because it is hosted by a DLL loaded inside our process

Similar issue to API monitoring using DLL injection



Demo



Evading Windows Defender

- Adding anti-emulation into our stager

