Independent Tests of
Anti-Virus Software

Shellcode Execution / Process Injection Certification

Bitdefender GravityZone Business Security Enterprise

TEST PERIOD: APRIL 2024

LAST REVISION: 21ST MAY 2024

WWW.AV-COMPARATIVES.ORG

Introduction

Every year, AV-Comparatives provides focus pen-tests, to which vendors can apply to get certified. This year we focus on "Shellcode Execution / Process Injection". Certification reports are published only for vendors who achieved the certification. Tested vendors received technical data about the test-cases and detailed feedback on how the products performed against the attacks in order to further improve their products.

Process injection is one of the most common techniques used by attackers (and hence red teams). By looking at the Process Injection[1] (T1055) Technique in the MITRE ATT&CK Framework, we can see that the technique currently includes around a dozen different sub-techniques, providing a lot of potential to be used by attackers (or red teams) in different contexts like initial access, defence evasion, privilege escalation etc.

Methodology

In this test we are interested in testing the prevention/detection capabilities of AV/EPP/EDR products regarding process injection/shellcode execution in the context of initial access. We want to evaluate how well the products perform when varied with different C2 frameworks/shellcode, different types of memory allocation methods, different types of shellcode execution, different types of APIs, different process injection methods, etc. We are also interested in checking how well the products perform when we vary the process into which we inject or execute our shellcode. Our aim is to evaluate whether various shellcode execution/process injection techniques are prevented or detected by an antivirus (AV), endpoint protection (EPP) or endpoint detection and response (EDR) solution.

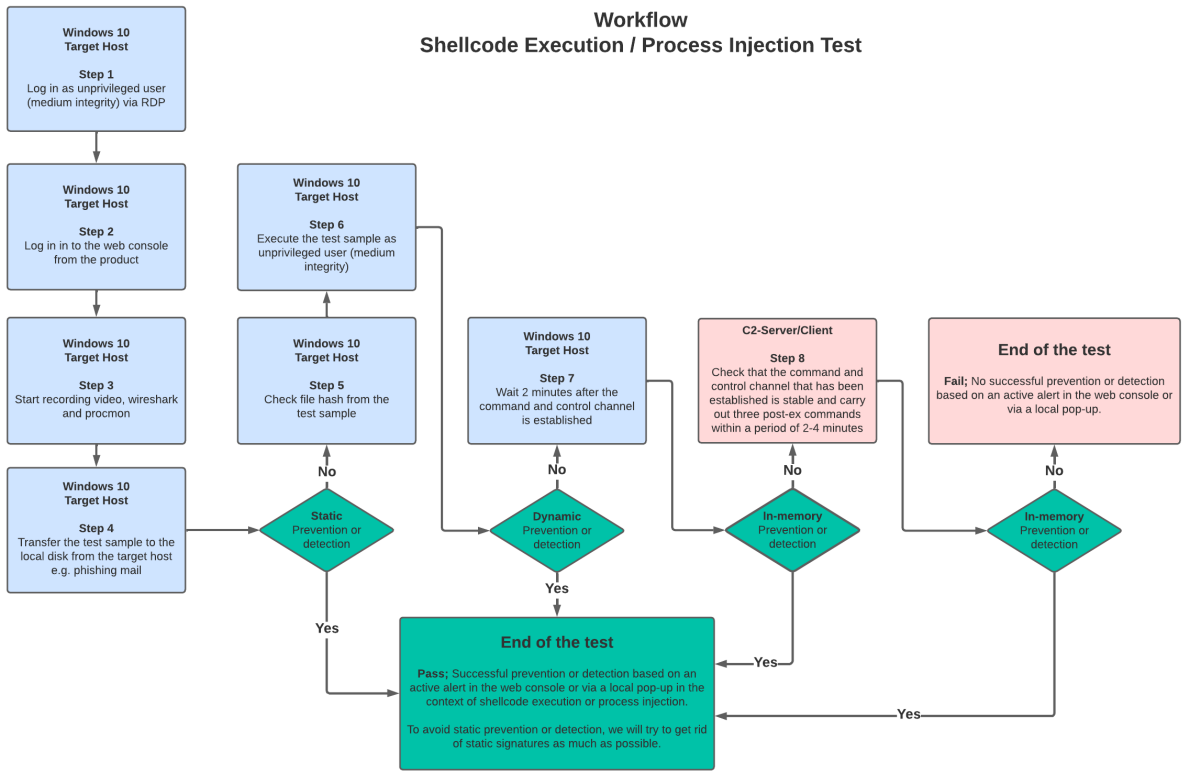
Below we have listed some possible variables that can be used to influence the creation of an evasive shellcode loader or process injection POC.

- **Execution/injection technique:** use of various techniques such as classic injection, early bird injection, and process hollowing.
- **Format/file type:** use of various file types like .exe, .dll, and .bin.
- **Frameworks/shellcode:** Use of different types of free and commercial command-and-control frameworks like Metasploit, Empire, Covenant, etc.
- **Self-injection/remote injection:** Variation as to whether the shellcode is executed locally in the same process (self-injection) or whether the shellcode is executed remotely in another process (remote injection).
- **Processes:** Variation of the process context in which the shellcode is executed (shellcode execution) or into which the shellcode is injected (process injection)

It should be noted that the Process Injection Test only tests one specific protection aspect (in contrast to e.g. AV-Comparatives' EPR[2] Test, which covers the entire attack chain). For the Process Injection Test, we use a fully patched and updated Windows 10 host. The tester logs on to Windows as a minimal user (Windows shell starts in medium integrity), and then executes the respective shellcode execution/process injection as an unprivileged user (medium integrity).

[1] https://attack.mitre.org/techniques/T1055/
[2] https://www.av-comparatives.org/enterprise/testmethod/endpoint-prevention-response-tests/

Workflow



Setup and configuration

The following setup is used to perform the tests:

- Windows 10 host, default configuration without any additional Windows hardening measures.
- All products must be configured so that prevention is active.
- Each product is to be configured by the manufacturer for the test, but once this configuration has been completed, it must not be further changed by the manufacturer or the tester during the entire test.
- No configuration is allowed that generally blocks the execution of files (executables, scripts, etc.).

Scope

- Test results focus on prevention and detection/active response capabilities in the event an attacker attempts to execute or inject shellcode for various malicious activities.

Out of Scope

The following points are not evaluated in the test and are therefore out of scope:

- Telemetry-based threat hunting from web console.
- Any action that could be taken after the shellcode execution or process injection (post-exploitation) completes successfully, where the product failed to prevent or detect the corresponding test case.

Tested Product

In this test, the following up-to-date and latest publicly available product was submitted by the vendor and tested in April 2024:

Bitdefender GravityZone Business Security Enterprise

Settings

In business environments, and with business products in general, it is usual for products to be configured by the system administrator in accordance with vendor's guidelines, and so we invited all vendors to configure their respective products. Each vendor had the opportunity to enable product-specific tamper protection settings (if not already activated by default), such as enabling uninstall protection, enabling tamper protection for settings, setting passwords, etc.

Below we have listed the product settings applied by the vendor. Setting changes that we consider were relevant for this test are highlighted in red.

Bitdefender: "Sandbox Analyzer" (for Applications, Documents, Scripts, Archives and Emails) enabled. "Analysis mode" set to "Monitoring". "Update ring" changed to "Fast ring". In "On-Access scanning", "Process memory Scan" was enabled, "Archive maximum size" was changed to "100MB" and "Archive maximum depth level" was set to "16". In "On-Execute", "Ransomware Mitigation" was enabled, as well as "AMSI Provider". All "AMSI Command-Line Scanner" settings enabled for "Fileless Attack Protection". In "HyperDetect", everything was set to "Aggressive". "LSASS protection" was set to "Block". "Vendor and product exclusions" was disabled. In "Network Protection", "Scan RDP" was enabled. "Web Traffic Scan" and "Email Traffic Scan" enabled for Incoming emails (POP3).

Please note that the results reached are valid only for the products tested with their respective settings. With other settings the Process Injection certification might not have been reached. Therefore, we urge readers to make sure that at least the settings marked in red are enabled/configured properly if they want to increase the Shellcode Execution / Process Injection protection of the product.

AV-Comparatives Process Injection Certification

To be approved by AV-Comparatives for Process Injection protection, a product must have successfully prevented **or** detected 2/3 (10/15) of the test cases, without false positives (without blocking legitimate applications).

Only products which were submitted for the Shellcode Execution / Process Injection Test, and which passed the test, are published. **Bitdefender GravityZone Business Security Enterprise** reached the certification requirements, i.e. successfully prevented or detected the Shellcode Execution / Process Injection attempts used in this test[3].

Successfully prevented or detected with active alert in the web console (or via local pop-up on the endpoint) at least 2/3 of the test cases in the context of Shellcode Execution / Process Injection.	✔

The following 15 test-cases have been tested (incl. false positive testing):

Process Injection Classic	**PASS**
Asynchronous Procedures Calls	**PASS**
Early Bird Asynchronous Procedures Calls	**PASS**
Thread Execution Hijacking	**PASS**
Process Hollowing	**PASS**
Transacted Hollowing	**PASS**
DLL Injection	**PASS**
Module Stomping	**PASS**
Function Stomping	**PASS**
Execution via Callback Function	**PASS**
Process Herpaderping	**PASS**
Threadless Injection	**PASS**
Thread Pools	**PASS**
TLS-Callbacks	**PASS**
Mapping Injection	**PASS**
False Positive Test	***PASS***

Key

Shellcode Execution / Process Injection blocked (with active alert)	**PASS**
Shellcode Execution / Process Injection was detected (with active alert), but not blocked.	**PASS***
Shellcode Execution / Process Injection hindered (without any alerts)	**PASS****
Shellcode Execution / Process Injection was neither blocked (with active alert) nor detected (with active alert)	**FAIL**
Legitimate applications in the False Positive Test were neither blocked, nor alerted on.	**PASS**
Legitimate applications in the False Positive Test were alerted on, but not blocked.	**PASS***
Legitimate applications in the False Positive Test were blocked.	**FAIL**

[3] Please note that the reached certification applies for the products tested with the settings specified on the previous page.

Copyright and Disclaimer

For more information about AV-Comparatives and the testing methodologies, please visit our website.

AV-Comparatives
(May 2024)