(S//NF) Network Operations Division Kernel-mode Execution Specification

(S//NF) NOD Specification 004: version 1

Classified By: 2343997

Derived From: CIA NSCG MET S-06

Declassify On: 25X1, 20641231

(U//FOUO) Contents

1. (U//FOUO) Overview

(S//OC/NF) This specification exists to provide a common interface for execution of code in kernel-space on Microsoft Windows hosts. A common interface allows the Sponsor to utilize multiple kernel-level tools while minimizing the unnecessary exposure of valuable local privilege escalation exploits and allows for replacement of LPEs without incurring a large-scale testing burden.

(S//OC/NF) Kernel execution involves a loader which has already gained execution in the Windows kernel and a payload which is kernel shellcode to be executed. Payload modules must consist of PIC shellcode for the appropriate architecture.

(S//OC/NF) The fact of generic CNE kernel-mode execution is classified SECRET//ORCON/NOFORN.

2. (U//FOUO) Loading and Invocation

2.1. (U//FOUO) The loader first gains execution through other means and begins executing operator-specified tasking.

2.2. (U//FOUO) The loader allocates a chunk of executable memory large enough to contain the payload. The Loader passes an argument from the operator to the module a `DWORD` value. The Loader then causes execution to begin at the first byte of the payload.

2.2.1. (U//FOUO) If possible the payload memory will be locked to prevent swapping to disk.

2.2.2. (U//FOUO) If the loader cannot allocate enough memory it will return an error to the operator.

2.2.3. (S//OC/NF) The payload will execute as a queued work item in the system process context.

2.2.4. (U//FOUO) Loaders will not verify that the architecture of the payload matches the architecture of the target computer.

2.3. (U//FOUO) The payload must return promptly in order to avoid deadlocking the Loader. After the payload has executed it may return a return code via the `EAX` or `RAX` register, as appropriate. This value will be displayed to the operator but has no inherent meaning within this specification.

2.4. (U//FOUO) After the payload has returned to the Loader the loader will zero and free the memory the payload occupied.

3. (U//FOUO) Different Architectures

(S//NF) Due to compiler limitations that would place an undue burden on every module developer to overcome this specification uses different calling conventions on `x86` and `x86_64` computers. Specifically Microsoft `__cdecl` is used on `x86` targets while the Microsoft 64 bit Calling Convention is used on `x86_64` targets.

(S//NF) For x86 targets the payload shellcode is called as if it were defined with the following prototype. One DWORD argument is always provided and it will default to the value of 0 if not otherwise specified by the operator.

```
DWORD __cdecl Shellcode(__in DWORD parameter);
```

(S//NF) For x86_64 targets the payload shellcode is called as if it were defined with the following prototype. Note that x86_64 payloads utilize the Microsoft 64 bit Calling Convention yet still expect and return DWORD values.

```
DWORD Shellcode(__in DWORD parameter);
```

Appendix A: (U//FOUO) Version History

(U//FOUO) Version 1: Initial publication