Module Structure

FROM BYOVD TO FULL SYSTEM EXPLOITATION

ADVANCED · KERNEL EXPLOITATION · 0X12 DARK DEV

THEORY

00 Introduction

Scope, ethical context, and the transition from kernel primitives to full system control.

01 Discovering Kernel Primitives

BYOVD fundamentals, vulnerable driver anatomy, IOCTL interfaces, and the GDRV arbitrary R/W primitive.

02 Exploiting Kernel Primitives

Building the read/write primitive from CVE-2018-19320 — IOCTL structure, DeviceIoControl wrappers, DriverOps.h.

TECHNIQUES

T1 Arbitrary Process Termination

Exploiting a driver IOCTL to terminate any process by PID — bypassing PPL and AV self-protection.

T2 From R/W Kernel to PPL Deactivation

Walking EPROCESS via ActiveProcessLinks and zeroing SignatureLevel, SectionSignatureLevel, and Protection.

T3 From R/W Kernel to DSE Deactivation

Patching g_CiOptions inside ci.dll to disable Driver Signature Enforcement and load unsigned drivers.

T4 From R/W Kernel to Privilege Escalation

Token stealing — copying the SYSTEM EPROCESS token into our own process for NT AUTHORITY\SYSTEM access.

T5 Persistent AV Evasion via Restart Loop

Dynamic PID resolution and continuous re-termination to keep AV processes down across SCM-triggered restarts.

T6 From R/W Kernel to DKOM Process Hiding

Direct Kernel Object Manipulation — unlinking an EPROCESS from ActiveProcessLinks to hide a process from userland enumeration.

→ **Exercise**

Challenge 1: EPROCESS walker + token theft LPE with dynamic offset resolution. Challenge 2: DSE disable + unsigned driver loader with g_CiOptions save/restore.

■ **Conclusions**

Kernel data as the attack surface — why HVCI and the vulnerable driver blocklist are the only controls that matter.