Ilmiy tadqiqotlar va ularning yechimlari jurnali
Aoka guvohnoma № 323759 ISSN: 3060-4818

ROOTKITLARNI TAHLIL QILISHDA WINDBG VOSITASIDAN FOYDALANISH VA O‘ZAK SATHIDAGI ZAIFLIKLARNI O‘RGANISH

Rohatov Sherzod Eldor o‘g‘li
Toshkent axborot texnologiyalari universiteti,
Kiberxavfsizlik fakulteti
4-bosqich talabasi, Toshkent, O‘zbekiston
E-mail: srohatov1@gmail.com

Olimov Iskandar Salimboyevich
Senior Lecturer, Department of
Cryptology Tashkent University of
Information Technologies named
after Muhammad al-Khwarizmi
Email: olimoviskandar89@gmail.com
https://doi.org/10.5281/zenodo.24193452

Annotatsiya

Maqolada Windows operatsion tizimida yadro darajasida ishlashi mumkin bo‘lgan rootkitlarni tahlil qilishda WinDbg vositasidan foydalanish metodikasi ko‘rib chiqiladi. Tadqiqotning maqsadi — xotira dump-fayllari, yuklangan drayverlar, modul va stek ma’lumotlarini tekshirishni mustaqil telemetriya manbalari bilan birlashtiruvchi izchil yondashuvni ishlab chiqish. Tadqiqotda Microsoft texnik hujjatlari va MITRE ATT&CK tahdid tasnifi tahlil qilindi, WinDbg, Sysmon, Code Integrity va Driver Verifier imkoniyatlari qiyoslandi. Natijada dalillarni saqlash, drayverlarni inventarizatsiya qilish, yadro holatini tekshirish va hodisalarni o‘zaro korrelyatsiya qilishdan iborat to‘rt bosqichli metodik model taklif etildi. Modelda bitta anomaliya rootkit mavjudligining isboti sifatida qabul qilinmaydi; xulosa mustaqil manbalar bilan tasdiqlanishi shart. Ish real zararli namuna bilan tajriba o‘tkazilganini da’vo qilmaydi va keyingi laboratoriya sinovlari uchun metodik asos yaratadi.

Kalit so‘zlar: rootkit, WinDbg, yadro rejimi, kernel-mode, drayver, xotira dump-fayli, Sysmon, Driver Verifier, HVCI, Code Integrity, raqamli kriminalistika, Windows xavfsizligi.

Аннотация

В статье рассматриваются возможности WinDbg при анализе подозрительной активности на уровне ядра Windows, включая исследование дампов памяти, проверку загруженных модулей и анализ стеков вызовов. Цель работы — систематизировать безопасную методику выявления признаков rootkit и уязвимых драйверов. Методология основана на анализе технической документации Microsoft и MITRE ATT&CK, сопоставлении диагностических индикаторов и разработке многоисточниковой схемы проверки. В качестве методического результата предложена последовательность действий: сохранение цифровых доказательств, инвентаризация драйверов, проверка целостности и подписи, сопоставление данных WinDbg с журналами Sysmon и событиями Code Integrity. Отдельно рассмотрены ограничения анализа: неполнота дампа, несовпадение символов, легитимные изменения кода и возможность ложноположительных выводов. Работа не заявляет о проведении эксперимента с реальным вредоносным образцом; предложенная схема предназначена для последующей проверки в изолированной виртуальной лаборатории.

Ключевые слова: руткит, WinDbg, ядро Windows, драйвер, дамп памяти, Sysmon, целостность кода, цифровая криминалистика.

Abstract

The article examines the use of WinDbg for analysing rootkits in Microsoft Windows, with particular attention to threats operating at kernel level. Rootkits may conceal processes, files, drivers, network connections and other operating-system objects by manipulating low-level mechanisms. The paper explains the distinction between user mode and kernel mode, outlines common kernel attack surfaces, and presents a defensive workflow for analysing kernel memory dumps and driver-loading artefacts with WinDbg. It also discusses complementary controls such as Sysmon, Driver Verifier, Code Integrity, Hypervisor-protected Code Integrity (HVCI), Secure Boot and Microsoft's vulnerable-driver blocklist. The proposed methodology is intended for controlled forensic and research environments and focuses on evidence collection, verification and mitigation rather than offensive implementation. No experimental measurements are claimed; the work is a methodological review based on vendor documentation and established threat-taxonomy sources.

Keywords: rootkit, WinDbg, kernel mode, Windows driver, memory dump, Sysmon, Driver Verifier, HVCI, Code Integrity, digital forensics.

1. Kirish

Operatsion tizim yadrosi kompyuter resurslarini boshqaruvchi eng imtiyozli dasturiy qatlam hisoblanadi. Jarayonlarni rejalashtirish, virtual xotirani boshqarish, qurilmalar bilan ishlash, fayl tizimi va tarmoq steki kabi muhim vazifalar yadro hamda yadro rejimida ishlovchi drayverlar orqali bajariladi. Ushbu qatlamdagi xatolik yoki ruxsatsiz o‘zgartirish butun tizimning maxfiyligi, yaxlitligi va mavjudligiga ta’sir qilishi mumkin. Shu sababli yadro darajasidagi zararli dasturlar, jumladan rootkitlar, axborot xavfsizligi nuqtayi nazaridan yuqori xavfli tahdidlar sirasiga kiradi.

Rootkit atamasi tizimda mavjud bo‘lgan zararli faoliyatni yashirishga qaratilgan texnikalar majmuasini anglatadi. MITRE ATT&CK tasnifida Rootkit T1014 texnikasi sifatida qayd etilgan: tajovuzkorlar dasturlar, fayllar, tarmoq ulanishlari, xizmatlar, drayverlar va boshqa komponentlarni yashirish uchun operatsion tizimning axborot beruvchi API chaqiruvlarini tutib qolishi yoki o‘zgartirishi mumkin [1]. Rootkit foydalanuvchi rejimida, yadro rejimida, yuklanish jarayonida yoki undan ham quyi qatlamlarda joylashishi mumkin. Ushbu maqolaning asosiy e’tibori Windows kernel-mode rootkitlarini tahlil qilishga qaratilgan.

Rootkitlarni aniqlashdagi asosiy muammo — kuzatuv manbasining o‘zi buzilgan bo‘lishi ehtimolidir. Masalan, foydalanuvchi rejimidagi vosita jarayonlar ro‘yxatini Windows API orqali olsa, yadro darajasida ushbu ro‘yxatni filtrlovchi komponent yashirilgan jarayonni ko‘rsatmasligi mumkin. Demak, tekshiruv faqat bitta API yoki bitta xavfsizlik mahsuloti natijasiga tayanmasligi, mustaqil manbalardan olingan dalillarni solishtirishi lozim.

WinDbg Microsoft tomonidan taqdim etilgan debugger bo‘lib, foydalanuvchi va yadro rejimlaridagi dasturiy komponentlarni tekshirish, xotira dump-fayllarini tahlil qilish, modul va stek holatini ko‘rish imkonini beradi [2]. WinDbg rootkitni avtomatik ravishda topib beradigan antivirus emas. Uning qiymati — tahlilchiga tizim holatini past darajada tekshirish, shubhali drayverlarni aniqlash va boshqa telemetriya bilan olingan taxminlarni tekshirish imkonini berishidadir.

Tadqiqotning maqsadi — WinDbg yordamida Windows yadro sathidagi shubhali faoliyatni tekshirishning metodik asoslarini ishlab chiqish, rootkitga xos indikatorlarni guruhlash va profilaktik himoya choralarini tizimlashtirish. Maqsadga erishish uchun quyidagi vazifalar belgilandi: rootkit tushunchasi va turlarini tahlil qilish; Windows yadro arxitekturasining xavfsizlikka taalluqli jihatlarini ko‘rib chiqish; WinDbg ish jarayonini bayon etish; drayver va xotira artefaktlarini tekshirish; aniqlash natijalarini baholash mezonlarini taklif qilish.

2. Mavzuga oid adabiyotlar tahlili

Ko‘rib chiqilgan manbalarda WinDbg, Sysmon va drayver xavfsizligi nazoratlari alohida yoritilgan. Ushbu maqolaning metodik hissasi ularni dalillarni saqlash, o‘zaro tekshirish va noto‘g‘ri ijobiy xulosalarni kamaytirishga yo‘naltirilgan yagona tahlil ketma-ketligida birlashtirishdan iborat.

Microsoft Vulnerable Driver Blocklist ma’lum zaif yoki zararli drayverlarning yuklanishini cheklashga qaratilgan. Biroq bloklash moslikka ta’sir qilishi mumkin; shu sababli siyosatni avval audit rejimida tekshirish va keyin bosqichma-bosqich joriy etish lozim [3]. Sysmon Event ID 6 drayver yuklanishi, fayl yo‘li, xesh va imzo haqidagi ma’lumotlarni qayd etib, WinDbg natijalarini mustaqil loglar bilan solishtirishga yordam beradi [4].

Drayver xavfsizligi bo‘yicha Microsoft tavsiyalari yadro rejimida bajariladigan kodning yuqori imtiyozini muhim xavf omili sifatida ko‘rsatadi. Xavfsiz drayver yaratishda bufer o‘lchamlarini tekshirish, user-mode’dan kelgan ko‘rsatkichlarni tekshirish, IOCTL interfeyslarini cheklash, HVCI bilan mos xotira ajratish va Driver Verifier orqali sinovdan o‘tkazish tavsiya etiladi [5, 6].

Microsoftning WinDbg hujjatlarida kernel-mode debugging, dump-fayllarni ochish, symbols yuklash va !analyze -v kabi diagnostika buyruqlaridan foydalanish tartibi bayon qilingan [2, 7]. Ushbu manbalar WinDbg’ni rootkitni avtomatik aniqlovchi antivirus sifatida emas, balki yadro holati va nosozlik dalillarini tekshiruvchi debugger sifatida qo‘llash zarurligini ko‘rsatadi.

Rootkitlarni aniqlash bo‘yicha yondashuvlar operatsion tizim ichki tuzilmalarini tekshirish va mustaqil telemetriya manbalaridan foydalanishga tayanadi. MITRE ATT&CK rootkitni tizimdagi zararli faoliyatni yashirishga qaratilgan texnika sifatida tasniflaydi; T1014 jarayonlar, fayllar, drayverlar va tarmoq artefaktlarini yashirish bilan bog‘liq [1]. MITRE aniqlash strategiyalari kernel va user-mode darajasidagi indikatorlarni birgalikda kuzatishni tavsiya qiladi [8].

2.1. Rootkitlarning tasnifi

Rootkitlarni joylashish qatlamiga ko‘ra bir necha turga ajratish mumkin. User-mode rootkitlar foydalanuvchi rejimidagi kutubxonalar, jarayonlar yoki API chaqiruvlariga ta’sir qiladi. Kernel-mode rootkitlar yadro rejimida ishlovchi drayver yoki boshqa komponent orqali tizim ma’lumotlarini o‘zgartirishga urinadi. Bootkitlar yuklanish zanjirining dastlabki bosqichlariga ta’sir ko‘rsatadi. Firmware yoki hypervisor darajasidagi rootkitlar esa operatsion tizimdan ham pastroq qatlamda joylashishi mumkin [1, 9].

Yadro rootkitlari ayniqsa xavfli, chunki kernel-mode kod tizim xotirasi va muhim obyektlariga keng imtiyoz bilan murojaat qila oladi. Microsoft hujjatlarida ham yadro rejimidagi drayverlar yuqori imtiyozga ega ekani, shu sababli drayverlarni raqamli imzolash va Code Integrity nazorati muhimligi ta’kidlanadi [5, 10].

2.2. Yadro sathidagi zaiflik tushunchasi

Yadro sathidagi zaiflik — yadro yoki drayverdagi dasturiy xato bo‘lib, undan foydalanish xotira buzilishi, imtiyozlarning oshishi, xizmat ko‘rsatishni rad etish yoki kod bajarilishiga olib kelishi mumkin. Keng tarqalgan sinflar qatoriga buffer overflow, use-after-free, double free, integer overflow, noto‘g‘ri pointer tekshiruvi, IRQL talablariga rioya qilmaslik, noto‘g‘ri sinxronlash va kirish huquqlarini yetarlicha tekshirmaslik kiradi.

Zaif drayver har doim rootkit degani emas. Biroq zaif yoki noto‘g‘ri himoyalangan drayver tajovuzkor uchun yadro imtiyozlariga erishish vositasiga aylanishi mumkin. Microsoftning zaif drayverlarni bloklash siyosati aynan ma’lum zaifliklarga ega uchinchi tomon drayverlarini yuklanishdan cheklashga qaratilgan [3].

3. Tadqiqot materiali va metodlari

Tadqiqot obyekti — Windows operatsion tizimida yadro rejimida bajariladigan komponentlar, xususan drayverlar, yadro xotirasi va ularning tizim yaxlitligiga ta’siridir. Tadqiqot predmeti esa rootkitga xos yashirish belgilari, yadro darajasidagi zaifliklar va ularni WinDbg hamda qo‘shimcha diagnostika vositalari yordamida aniqlash metodlaridan iborat.

Ish metodologiyasi hujjatlarni tahlil qilish, tahdidlarni tasniflash, diagnostik indikatorlarni taqqoslash va laboratoriya tekshiruvining taklif etilgan ketma-ketligini ishlab chiqishga asoslanadi. Microsoft Learn hujjatlari WinDbg, kernel dump va drayverlarni tekshirish bo‘yicha texnik asos sifatida olindi [2, 5, 7]. MITRE ATT&CK esa rootkit faoliyatini tahdid modeli va aniqlash strategiyalari nuqtayi nazaridan tasniflash uchun qo‘llanildi [1, 8].

Maqolada real zararli dastur bilan tajriba o‘tkazilgani yoki muayyan tizimda rootkit topilgani da’vo qilinmaydi. Keltirilgan buyruqlar diagnostika va kuzatuv maqsadida berilgan. Yadro xotirasiga yozish, noma’lum drayverni yuklash yoki tizim himoyasini o‘chirish tavsiya etilmaydi. Amaliy tekshiruvlar snapshot olish mumkin bo‘lgan virtual mashinada, alohida test muhiti va qayta tiklash nuqtasi bilan bajarilishi kerak.

Tahlil birligi sifatida quyidagi artefaktlar belgilandi: (1) WinDbg orqali olingan yuklangan kernel modullari ro‘yxati; (2) dump ichidagi bug check va stack ma’lumotlari; (3) Sysmon Event ID 6 yozuvlari; (4) drayverning raqamli imzosi, versiyasi va SHA-256 xeshi; (5) Code Integrity va Windows xavfsizlik siyosati hodisalari. Har bir artefakt bo‘yicha kuzatuv hamda uning muqobil izohi alohida qayd etiladi.

Taqqoslash usuli quyidagicha belgilandi: WinDbg modul ro‘yxatidagi uchinchi tomon drayverlari Sysmon yozuvlari bilan vaqt va fayl nomi bo‘yicha solishtiriladi; drayver xeshi va imzosi ishonchli manbadagi versiya bilan tekshiriladi; stack trace’da uchraydigan modul nomlari yuklangan modullar ro‘yxati bilan moslashtiriladi. Natija faqat kamida ikki mustaqil manba bilan tasdiqlanganda yuqori ishonchli indikator sifatida baholanadi.

Cheklov: ushbu ish konseptual-metodik tadqiqot bo‘lib, real Windows hostida tajriba o‘tkazilmagan, aniqlash aniqligi, noto‘g‘ri ijobiy holatlar yoki ishlash tezligi bo‘yicha statistik ko‘rsatkichlar olinmagan. Shu sababli keyingi bo‘limdagi natijalar amaliy tajriba natijasi emas, balki manbalar tahlilidan chiqarilgan metodik model natijasi sifatida taqdim etiladi.

4. Windows yadro arxitekturasi va hujum yuzasi

Windows arxitekturasi odatda foydalanuvchi rejimi (user mode) va yadro rejimi (kernel mode) sifatida ajratiladi. Foydalanuvchi rejimidagi dasturlar o‘z virtual manzil makonida ishlaydi va operatsion tizim resurslariga tizim chaqiruvlari hamda hujjatlashtirilgan API orqali murojaat qiladi. Yadro rejimidagi kod esa operatsion tizimning asosiy qismlari va ko‘plab qurilma drayverlarini o‘z ichiga oladi. Microsoft ta’rifiga ko‘ra, yadro rejimidagi kod boshqa jarayonlarning xotirasiga va tizim resurslariga foydalanuvchi rejimiga nisbatan kengroq kirish huquqiga ega [10, 11].

Windows yadro muhitida NT kernel, HAL, Executive komponentlari, xotira menejeri, I/O menejeri, obyekt menejeri, jarayon va oqim boshqaruvi hamda drayverlar o‘zaro hamkorlik qiladi. Drayverlar apparat qurilmalar bilan ishlashdan tashqari, fayl tizimi filtrlari, tarmoq filtrlari va virtual qurilmalar kabi funksiyalarni ham bajarishi mumkin. Shu sababli drayver interfeyslari va IOCTL so‘rovlari xavfsizlik tekshiruvida alohida e’tiborga loyiq.

4.1. Rootkitlar foydalanishi mumkin bo‘lgan mexanizmlar

Rootkitlar turli yashirish usullaridan foydalanishi mumkin. API hooking foydalanuvchi rejimidagi funksiyalar natijasini o‘zgartiradi. Kernel callback yoki filtrlash mexanizmlarini suiiste’mol qilish jarayon, registr, fayl tizimi yoki tarmoq hodisalarini kuzatish va ayrim hollarda yashirish imkonini beradi. Kernel data structure tampering esa yadro boshqaruv tuzilmalaridagi ma’lumotlarni o‘zgartirishga qaratiladi. Bunday o‘zgarishlarni aniqlash uchun faqat yuqori darajadagi ro‘yxatlarni emas, balki yadro holati, drayverlar va mustaqil log manbalarini ham solishtirish talab etiladi.

Inline hooking yoki kod segmentini o‘zgartirish holatlarida modulning xotiradagi kodi diskdagi ishonchli nusxadan farq qilishi mumkin. Biroq xotiradagi farq har doim zararli faoliyatni anglatmaydi: hotpatch, xavfsizlik mahsulotlari, virtualizatsiya yoki operatsion tizim yangilanishlari ham farq keltirib chiqarishi mumkin. Shu sababli har qanday anomaliya versiya, imzo, modul manzili, yuklanish vaqti va tizim konfiguratsiyasi bilan birga talqin qilinadi.

4.2. Drayverlar va ishonch zanjiri

Drayverning raqamli imzosi uning yaxlitligi va nashriyotchi identifikatsiyasini tekshirishga yordam beradi. Microsoft drayverlarni imzolash siyosatida 64-bit Windows versiyalarida kernel-mode drayverlarni yuklash uchun tegishli imzo talablari mavjudligini bayon qiladi [5]. Biroq imzolangan drayver avtomatik ravishda xavfsiz degani emas: haqiqiy sertifikat bilan imzolangan, ammo zaif drayver ham hujumda ishlatilishi mumkin. Shuning uchun imzo tekshiruvi hash, versiya, ishlab chiqaruvchi, fayl joylashuvi, yuklanish konteksti va zaif drayverlar ro‘yxati bilan birgalikda baholanishi lozim.

Windows 11 muhitida Microsoft Vulnerable Driver Blocklist ma’lum zaif drayverlarni bloklashga xizmat qiladi. Microsoft hujjatlarida ushbu ro‘yxat HVCI, Smart App Control yoki S mode yoqilgan holatlarda qo‘llanishi, shuningdek ro‘yxatning yangilanib borishi qayd etilgan [3]. Bu mexanizm rootkitni aniqlashning o‘rnini bosmaydi, lekin ma’lum zaif drayverlardan foydalanish xavfini kamaytiradi.

5. WinDbg vositasining imkoniyatlari

WinDbg — Windows uchun Microsoft debuggeri bo‘lib, yadro rejimi va foydalanuvchi rejimidagi jarayonlarni tahlil qilish, dump-fayllarni ochish, modul va stek ma’lumotlarini ko‘rish, xotira manzillarini tekshirish hamda debugger kengaytmalaridan foydalanish imkonini beradi [2]. Kernel debugging odatda ikkita kompyuter — host va target — orqali tashkil etiladi. Debugger host kompyuterda ishlaydi, tekshirilayotgan Windows esa target kompyuterda bajariladi. Microsoft hujjatlarida Ethernet, USB 3.0 va serial ulanishlar kabi usullar ko‘rsatilgan [2].

WinDbg yordamida rootkit tahlilida ikki asosiy yondashuv mavjud: jonli kernel debugging va crash dump tahlili. Jonli debugging tizim ishlayotgan vaqtda yadro holatini ko‘rish imkonini beradi, biroq noto‘g‘ri buyruq yoki breakpoint tizim faoliyatiga ta’sir qilishi mumkin. Dump tahlili esa oldindan olingan xotira tasvirini tekshirishga imkon beradi va dastlabki kriminalistik tahlil uchun ko‘pincha xavfsizroq hisoblanadi. Dump tarkibida faqat kernel memory yoki kengroq xotira segmentlari bo‘lishi mumkin; tahlil imkoniyati dump turiga bog‘liq.

5.1. WinDbg muhitini tayyorlash

Tahlil uchun WinDbg Microsoft Store yoki Windows SDK tarkibidagi Debugging Tools for Windows orqali o‘rnatiladi. Dump-faylni ochishdan oldin operatsion tizim versiyasi, build raqami, arxitektura, dump olingan vaqt va faylning kriptografik xeshi qayd etilishi kerak. Tahlilchi asl faylni o‘zgartirmasdan, ishchi nusxa bilan ishlashi tavsiya etiladi.

WinDbg ishga tushirilgach, Microsoft Symbol Server orqali mos PDB belgilarini yuklash sozlanadi. Belgilar (symbols) yadro funksiyalari, modul nomlari va ma’lumotlar tuzilmalarini to‘g‘ri talqin qilish uchun zarur. Belgilar noto‘g‘ri yoki mos bo‘lmasa, buyruqlar natijasi chalg‘ituvchi bo‘lishi mumkin. Microsoft kernel debugging qo‘llanmasida .sympath va .reload buyruqlari yordamida belgilar yo‘lini sozlash ko‘rsatilgan [2].

```
.sympath srv*c:\symbols*https://msdl.microsoft.com/download/symbols
.reload
lm
!analyze -v
.bugcheck
!process 0 0
kv
```

Yuqoridagi buyruqlar diagnostika uchun mo‘ljallangan. lm yuklangan modullar ro‘yxatini ko‘rsatadi; !analyze -v dump yoki bug check bo‘yicha avtomatik tahlil beradi; .bugcheck xato kodi va parametrlarini chiqaradi; !process 0 0 jarayonlar haqidagi yadro ma’lumotlarini ko‘rsatadi; kv esa joriy kontekstda stek izini chiqaradi. Buyruqlar natijasi dump turi va debugger kontekstiga qarab farq qilishi mumkin [7].

5.2. Dump-faylni ochish va dastlabki tahlil

Kernel dump fayli odatda .dmp yoki .mdmp kengaytmasiga ega bo‘ladi. Uni WinDbg orqali File → Open Crash Dump menyusidan yoki buyruq satrida quyidagi sintaksis orqali ochish mumkin [7]:

```
windbg -y <SymbolPath> -i <ImagePath> -z <DumpFileName>
```

Dump ochilgandan so‘ng birinchi navbatda !analyze -v buyrug‘i bajariladi. Natijada bug check kodi, ehtimoliy sabab, jarayon konteksti va stack trace kabi ma’lumotlar ko‘rsatilishi mumkin. Keyin .bugcheck orqali xato parametrlari alohida qayd etiladi. Agar xato drayver bilan bog‘liq bo‘lsa, modul ro‘yxati (lm), yuklangan modulning versiyasi va xotiradagi manzil oralig‘i tekshiriladi. Microsoft kernel dump tahlili bo‘yicha aynan !analyze, .bugcheck, lm, !process va boshqa kengaytmalarni dastlabki tekshiruvda qo‘llashni tavsiya qiladi [7].

6. Rootkitga shubhali holatlarni WinDbg orqali tekshirish metodikasi

Rootkitni tahlil qilishda bitta indikator asosida xulosa chiqarish metodik jihatdan noto‘g‘ri. Tavsiya etiladigan yondashuv — bir nechta mustaqil dalil manbalarini yig‘ish, ular o‘rtasidagi nomuvofiqlikni aniqlash va har bir anomaliyani muqobil izohlar bilan tekshirish. Quyidagi metodika incident response va raqamli kriminalistika jarayoniga moslashtirilgan.

6.1. 1-bosqich: dalillarni saqlash

Dastlab tizim identifikatorlari, vaqt zonasi, Windows build raqami, so‘nggi yangilanishlar, foydalanuvchi konteksti va hodisa aniqlangan vaqt qayd etiladi. Agar tizim faol kompрometatsiya ostida bo‘lishi mumkin bo‘lsa, uni qayta yuklash yoki tozalashdan oldin tashkilotning incident response tartibiga muvofiq volatile evidence — operativ xotira, faol ulanishlar va jarayon holati — saqlanishi ko‘rib chiqiladi [12]. Xotira dump-faylini olish tashkilot siyosati va maxfiylik talablariga muvofiq amalga oshiriladi.

Dump-fayl uchun SHA-256 xesh hisoblanadi, fayl nusxalari alohida saqlanadi va kim, qachon, qaysi vosita bilan olgani qayd etiladi. Bu chain of custody, ya’ni dalilning saqlanish zanjirini ta’minlashga xizmat qiladi. Dump tarkibida parollar, tokenlar, shaxsiy ma’lumotlar yoki maxfiy kalitlar bo‘lishi mumkinligi sababli unga kirish cheklanishi zarur.

6.2. 2-bosqich: modul va drayverlarni inventarizatsiya qilish

WinDbg’da lm buyrug‘i orqali yuklangan modullar ro‘yxati olinadi. Har bir uchinchi tomon drayveri bo‘yicha nomi, manzil oralig‘i, ishlab chiqaruvchi, versiya, fayl yo‘li, imzo holati va SHA-256 xeshi tekshiriladi. Shubhali belgilar quyidagilar bo‘lishi mumkin: noma’lum yoki kutilmagan modul; tizim drayverlari katalogidan tashqarida joylashgan kernel drayveri; imzo holatining noto‘g‘ri bo‘lishi; tizim vazifasiga mos kelmaydigan nom; kutilmagan yuklanish vaqti; ma’lum zaif drayver versiyasi.

Biroq modul nomining g‘alati ko‘rinishi o‘zi bilan zararli faoliyatni isbotlamaydi. Ayrim apparat ishlab chiqaruvchilari drayverlari kam tanilgan nomlarga ega bo‘lishi mumkin. Har bir topilma ishlab chiqaruvchining rasmiy manbasi, fayl xeshi va sertifikat zanjiri bilan solishtirilishi kerak.

6.3. 3-bosqich: jarayonlar va stek izlarini tekshirish

!process 0 0 buyrug‘i jarayonlar ro‘yxati bo‘yicha yadro ma’lumotlarini ko‘rishga yordam beradi. Natija Event Log, EDR telemetriyasi yoki boshqa mustaqil manbalardagi jarayonlar bilan solishtiriladi. Agar bir manbada mavjud bo‘lgan jarayon boshqa manbada ko‘rinmasa, bu yashirish ehtimolini tekshirish uchun signal bo‘lishi mumkin. Lekin jarayonning tugashi, dump olingan vaqt, kontekst cheklovlari yoki dump turining yetarli emasligi ham tafovut keltirib chiqarishi mumkin.

Stack trace tahlili shubhali modulning qaysi kontekstda chaqirilganini tushunishga yordam beradi. kv yoki k buyruqlari stek ramkalarini ko‘rsatadi. Agar uchinchi tomon drayveri kutilmagan chaqiruv zanjirida paydo bo‘lsa, uning vazifasi, imzosi va tegishli hodisalar tekshiriladi. Stack trace’ni noto‘g‘ri talqin qilmaslik uchun mos symbols yuklanganiga ishonch hosil qilish lozim.

6.4. 4-bosqich: kod va xotira yaxlitligini baholash

Shubhali modulning xotiradagi kodi ishonchli disk nusxasi bilan taqqoslanishi mumkin. Bunda modulning aynan bir xil build va arxitekturaga tegishli nusxasi olinishi, relocations va dinamik

o‘zgarishlar hisobga olinishi zarur. Kod segmentidagi farq rootkit belgisi bo‘lishi mumkin, lekin patch, hotpatch yoki xavfsizlik mahsuloti injeksiyasi kabi qonuniy sabablar ham mavjud. Shuning uchun farqning joyi, hajmi, bajariladigan sahifa atributlari va tegishli modul konteksti birgalikda tahlil qilinadi.

Kernel memory tahlili yuqori malaka talab qiladi. WinDbg xotirani o‘qish va ayrim holatlarda yozish imkonini beradi, ammo noto‘g‘ri manzilga yozish tizimni ishdan chiqarishi mumkinligi Microsoft hujjatlarida alohida ogohlantirilgan [2]. Shu bois ushbu maqolada xotirani o‘zgartiruvchi buyruqlar amaliy ko‘rsatma sifatida berilmaydi; tahlil faqat o‘qish va hujjatlashtirishga qaratiladi.

7. Qo‘shimcha telemetriya va tahlil vositalari

WinDbg yadro holatini chuqur tekshirish imkonini bersa-da, rootkitni aniqlash jarayonida boshqa vositalar bilan birga qo‘llanganda samaraliroq bo‘ladi. Tizim loglari, drayver yuklanish hodisalari, Code Integrity yozuvlari, antivirus/EDR signallari va fayl tizimi artefaktlari WinDbg natijalarini kontekst bilan ta’minlaydi.

7.1. Sysmon va Driver Load hodisalari

Microsoft Sysmon Windows tizimida muhim xavfsizlik hodisalarini qayd etuvchi Sysinternals vositasidir. Sysmon Event ID 6 drayver yuklanganini, uning fayl manzili, xesh va imzo haqidagi ma’lumotlarni qayd etadi [4]. Bu hodisa rootkitga aloqador drayverlar yuklanishini aniqlashda foydali, chunki kernel driver load hodisalari odatda nisbatan kam uchraydi va yuqori qiymatli telemetriya hisoblanadi [4].

Sysmon Event ID 6 ni tahlil qilganda quyidagilar tekshiriladi: drayver qaysi vaqtda yuklangan; fayl qayerdan ishga tushgan; raqamli imzo mavjudmi; imzo holati validmi; xesh ma’lum zararli yoki zaif drayverlar bazasiga mos keladimi; yuklanishdan oldin yangi servis yaratilganmi; shu vaqtda boshqa shubhali jarayon yoki tarmoq hodisasi kuzatilganmi. MITRE ATT&CK rootkitni aniqlash strategiyasida ruxsatsiz yoki noodatiy drayver yuklanishi, yashirilgan servislar va boot komponentlaridagi o‘zgarishlarni muhim indikatorlar sifatida keltiradi [8].

7.2. Driver Verifier

Driver Verifier — drayverlarning noto‘g‘ri xatti-harakatlarini aniqlashga mo‘ljallangan Windows diagnostika vositasi. U maxsus pool, IRQL tekshiruvi, I/O verification, deadlock detection, DMA tekshiruvlari va boshqa qoidalarni qo‘llashi mumkin [13]. Microsoft hujjatlarida Driver Verifier ayrim xatolarni aniqlash uchun tizimni bug check holatiga olib kelishi va dump yaratishi mumkinligi ko‘rsatilgan [13].

Driver Verifier asosan drayver ishlab chiqish va testlashda ishlatiladi; uni ishlab turgan muhim server yoki foydalanuvchi kompyuterida ehtiyotsiz yoqish tavsiya etilmaydi. Noto‘g‘ri konfiguratsiya tizim barqarorligiga salbiy ta’sir qilishi mumkin. Rootkit tekshiruvlarida undan faqat izolyatsiyalangan laboratoriya muhitida va tiklash rejasi mavjud bo‘lganda foydalanish maqsadga muvofiq.

7.3. Code Integrity, HVCI va Secure Boot

Code Integrity yadroga yuklanadigan drayverlarning ishonchliligini tekshirishga xizmat qiladi. Raqamli imzo va drayver siyosati yadroga noma’lum kod yuklanish xavfini kamaytiradi [5]. HVCI yoki Memory Integrity esa virtualizatsiyaga asoslangan xavfsizlik muhitidan foydalanib, yadro rejimida bajariladigan kodning yaxlitligini kuchaytiradi [14, 15]. Secure Boot esa yuklanish zanjirida ishonchli komponentlarni tekshirishga yordam beradi.

Microsoft Vulnerable Driver Blocklist ma’lum zaif drayverlarni bloklash orqali BYOVD (Bring Your Own Vulnerable Driver) uslubidagi hujumlar xavfini pasaytiradi [3]. Biroq blok-ro‘yxat ham mutlaq himoya emas: yangi aniqlangan zaifliklar, noto‘g‘ri konfiguratsiya yoki moslik sababli ayrim holatlar chetda qolishi mumkin. Shuning uchun yangilanishlarni o‘z vaqtida o‘rnatish, drayverlarni inventarizatsiya qilish va monitoringni davom ettirish zarur.

8. Rootkit indikatorlarini baholash modeli

Tahlil natijalarini izchil baholash uchun maqolada sifatga asoslangan indikatorlar modeli taklif etiladi. Model har bir indikatorni alohida ball bilan baholashni emas, balki uning ishonchliligi, mustaqil

tasdiqlanishi va tizim kontekstiga mosligini qayd etishni nazarda tutadi. Bu yondashuv yolg‘on ijobiy xulosalarni kamaytirishga yordam beradi. Indikator guruhlari 1-jadvalda keltirilgan.

1-jadval. Rootkit indikator guruhlari, kuzatiladigan belgilar va tekshirish manbalari

Indikator guruhi	Kuzatiladigan belgi	Tekshirish manbasi
Drayver yaxlitligi	Noma’lum, imzosiz yoki zaif versiyadagi drayver	Sysmon EID 6, Code Integrity, fayl xeshi
Modullar ro‘yxati	Kutilmagan modul yoki shubhali yuklanish manzili	WinDbg: lm; diskdagi drayver inventari
Jarayon ko‘rinishi	Mustaqil manbalardagi jarayonlar ro‘yxati tafovuti	WinDbg: !process; EDR/Sysmon
Kod yaxlitligi	Xotiradagi kod va ishonchli nusxa o‘rtasida izohlanmagan farq	WinDbg, hash, imzo va versiya tekshiruvi
Yuklanish zanjiri	Boot konfiguratsiyasi yoki servis yozuvlaridagi kutilmagan o‘zgarish	Secure Boot holati, Event Log, autoruns
Barqarorlik xatolari	Takroriy bug check, pool yoki IRQL xatolari	!analyze -v, Driver Verifier

Indikatorlar quyidagi toifalarga ajratilishi mumkin: past — bitta noaniq anomaliya; o‘rta — bir nechta mos keluvchi, lekin to‘liq tasdiqlanmagan belgi; yuqori — mustaqil manbalarda tasdiqlangan, izohlanmagan yadro o‘zgarishi yoki ma’lum zararli komponent bilan moslik. Bu tasnif avtomatik skoring algoritmi emas, balki analitik qaror qabul qilish uchun tavsiya etilgan sifat mezonidir.

8.1. Dalillarni o‘zaro tekshirish

Masalan, Sysmon Event ID 6 da kutilmagan drayver yuklangani qayd etildi deylik. Bu hodisa mustaqil ravishda rootkit borligini isbotlamaydi. Tahlilchi drayverning imzo holatini, xeshini, versiyasini, yuklanish vaqtini, servis yaratilishi hodisasini va WinDbg modul ro‘yxatini solishtirishi kerak. Agar drayver rasmiy ishlab chiqaruvchiga tegishli bo‘lsa, lekin zaif versiyada bo‘lsa, bu rootkit emas, ammo ekspluatatsiya qilinishi mumkin bo‘lgan xavfli komponent bo‘lishi mumkin.

Agar jarayonlar ro‘yxatida nomuvofiqlik aniqlansa, dump olingan vaqt va jarayonning hayot sikli tekshiriladi. Jarayon dumpdan oldin tugagan bo‘lishi, ayrim dump turlarida kerakli sahifalar mavjud bo‘lmasligi yoki debugger konteksti yetarli bo‘lmasligi mumkin. Shu sababli har bir xulosa uchun dalil, muqobil izoh va ishonch darajasi hujjatlashtiriladi.

8.2. Xatolik va cheklovlar

WinDbg tahlili mutaxassis tajribasiga, to‘g‘ri symbols tanlanganiga, dump sifati va tizim versiyasiga bog‘liq. Rootkit yadrodagi ma’lumotlarni manipulyatsiya qilgan bo‘lsa, ayrim ma’lumotlar dump ichida ham noto‘g‘ri ko‘rinishi mumkin. Shu bois xotira tahlili disk forensikasi, boot o‘lchovlari, loglar va endpoint telemetriyasi bilan birga bajarilishi kerak. Tahlilchi har doim normal tizim xatti-harakatlari va xavfsizlik dasturlari kiritadigan o‘zgarishlarni hisobga olishi lozim.

9. Laboratoriya ishini tashkil etish bo‘yicha tavsiyalar

Rootkit va yadro zaifliklarini o‘rganish real qurilmada emas, alohida virtual laboratoriyada bajarilishi maqsadga muvofiq. Laboratoriya host kompyuter, Windows target virtual mashinasi, WinDbg, Windows SDK/WDK, Sysmon va kerak bo‘lsa Driver Verifier vositalaridan tashkil topishi mumkin. Target mashinada shaxsiy yoki ishlab chiqarish ma’lumotlari saqlanmasligi, tarmoqdan ajratish yoki cheklangan virtual tarmoqdan foydalanish, snapshotlar yaratish va tajriba tugagach muhitni tiklash zarur.

1. Toza Windows virtual mashinasini o‘rnatish, barcha xavfsizlik yangilanishlarini qo‘llash va VM snapshot yaratish.

2. WinDbg va Sysmon o‘rnatish; tizim build raqami, drayverlar ro‘yxati va boshlang‘ich loglarni qayd etish.

3. Microsoft hujjatlariga muvofiq kernel debugging yoki dump tahlili muhitini sozlash; symbols yo‘lini tekshirish.

4. Faqat ishonchli, Microsoft tomonidan taqdim etilgan test materiallari yoki maxsus o‘quv drayverlari bilan diagnostika mashqlarini bajarish.

5. Sysmon Event ID 6, Code Integrity hodisalari va WinDbg modul ro‘yxatini o‘zaro taqqoslash.

6. Natijalarni jadvalga kiritish: vaqt, indikator, manba, kuzatuv, muqobil izoh va xulosa.

7. VM holatini snapshotdan tiklash va laboratoriya hisobotini yakunlash.

10. Natijalar: ko‘p manbali tahlil modeli

Manbalar tahlili asosida rootkitga shubhali holatlarni baholash uchun to‘rt qatlamli model ishlab chiqildi: (1) dalillarni yig‘ish va yaxlitligini saqlash; (2) drayver va modul inventarizatsiyasi; (3) yadro holatini WinDbg orqali tekshirish; (4) mustaqil telemetriya bilan korrelyatsiya va xulosa chiqarish. Modelning asosiy talabi — bitta anomaliyani rootkit mavjudligining yetarli isboti deb qabul qilmaslik. Indikatorlarni tekshirish va talqin qilish matritsasi 2-jadvalda keltirilgan.

2-jadval. Rootkitga shubhali indikatorlarni tekshirish va talqin qilish matritsasi

Indikator	Asosiy tekshiruv	Mustaqil tasdiq	Talqin cheklovi
Kutilmagan kernel moduli	WinDbg: lm; modul nomi va manzil oralig‘i	Sysmon EID 6, drayver fayli, imzo va xesh	Noma’lum nomning o‘zi zararli ekanini isbotlamaydi
Drayver yuklanishi	Yuklanish vaqti, yo‘li, versiyasi	Sysmon, Code Integrity, servis konfiguratsiyasi	Qonuniy yangilanish yoki apparat drayveri bo‘lishi mumkin
Jarayonlar ro‘yxatidagi tafovut	WinDbg: !process 0 0	EDR/Sysmon va hodisa vaqti	Dump turi yoki jarayonning tugashi sababli farq bo‘lishi mumkin
Shubhali stack/modul chaqiruvi	WinDbg: kv yoki k; symbols holati	Modul imzosi, versiyasi, ishlab chiqaruvchi hujjati	Noto‘g‘ri symbols stack talqinini buzishi mumkin
Kod yaxlitligi farqi	Bir xil build’dagi disk nusxasi bilan solishtirish	Hash, imzo, Code Integrity hodisalari	Hotpatch va xavfsizlik dasturi qonuniy farq keltirishi mumkin
Zaif drayver indikatori	Versiya va xeshni blocklist bilan solishtirish	Microsoft blocklist va ishlab chiqaruvchi advisory	Blocklist to‘liq emas; moslik xavfi mavjud

Model bo‘yicha xulosa darajasi uch bosqichda belgilanadi: past ishonch — bitta noaniq indikator; o‘rta ishonch — bir nechta mos keluvchi, ammo mustaqil tasdiqlanmagan indikator; yuqori ishonch — bir-biridan mustaqil manbalarda tasdiqlangan va qonuniy izoh bilan tushuntirilmagan yadro darajasidagi anomaliya. Bu darajalar ekspert bahosini standartlashtirishga xizmat qiladi, ammo statistik ehtimollik sifatida talqin qilinmaydi.

Tahlil hisobotining minimal tarkibi quyidagicha belgilandi: artefakt identifikatori; olingan vaqt; manba va vosita versiyasi; buyruq yoki log filtri; kuzatuv; hash/imzo holati; muqobil izoh; ishonch darajasi; tavsiya etilgan keyingi tekshiruv. Bunday qaydlar tekshiruvni takrorlash va dalillar zanjirini saqlash imkonini beradi.

11. Muhokama

Tahlil shuni ko‘rsatadiki, WinDbg rootkitlarni tekshirishda muhim, ammo yagona bo‘lmagan vositadir. Uning asosiy afzalligi — yadro darajasidagi ma’lumotlarni ko‘rish, dump-faylni tizimli tekshirish, modul va stek holatini tahlil qilish imkoniyatidir. Shu bilan birga, WinDbg avtomatik zararli dastur klassifikatori emas va natijalarni to‘g‘ri talqin qilish uchun Windows ichki arxitekturasi, xotira boshqaruvi hamda drayver modeli bo‘yicha bilim talab etadi.

Rootkitlar yashirishga qaratilganligi sababli tahlilni faqat operatsion tizimning o‘z API natijalariga asoslash zaif yondashuv hisoblanadi. Mustaqil telemetriya — masalan, Sysmon Event ID 6 — drayver yuklanishini qayd etadi va WinDbg orqali olingan modul ro‘yxati bilan solishtirish imkonini beradi [4]. MITRE ATT&CK ham rootkitni aniqlashda ruxsatsiz drayver yuklanishi, yashirilgan servislar va boot komponentlaridagi o‘zgarishlarni birgalikda kuzatishni tavsiya qiladi [8].

Yadro zaifliklarini kamaytirishda preventiv nazorat muhim. Drayverlarni faqat ishonchli manbalardan o‘rnatish, raqamli imzo talablarini saqlash, HVCI va Secure Boot imkoniyatlarini mos qurilmalarda yoqish, zaif drayverlar blok-ro‘yxatini yangilab borish hamda tizim yangilanishlarini kechiktirmaslik kerak [3, 5, 15]. Biroq himoya sozlamalarini o‘zgartirishdan oldin tashkilotdagi apparat va dasturiy ta’minot mosligi sinovdan o‘tkazilishi lozim.

Tadqiqotning cheklovi shundan iboratki, unda real zararli rootkit namunasi bilan laboratoriya tajribasi va miqdoriy samaradorlik ko‘rsatkichlari keltirilmagan. Shu sababli maqoladagi model amaliy sinovdan o‘tgan aniqlash algoritmi sifatida emas, balki manbalarga asoslangan metodik tavsiya sifatida talqin qilinishi kerak. Kelgusidagi tadqiqotlarda xavfsiz test drayverlari, oldindan tayyorlangan dump-fayllar va sintetik Sysmon hodisalari yordamida indikatorlarning aniqligi, noto‘g‘ri ijobiy holatlar va tahlil vaqtini o‘lchash mumkin.

12. Xulosa va amaliy tavsiyalar

Rootkitlar Windows operatsion tizimining ishonch modeliga tahdid soluvchi murakkab zararli komponentlar bo‘lib, yadro rejimida ishlaganda ularning ta’sir doirasi kengayadi. Ular jarayonlar, fayllar, drayverlar va tarmoq obyektlarini yashirish uchun past darajadagi mexanizmlardan foydalanishi mumkin. Shu sababli rootkitlarni aniqlashda oddiy foydalanuvchi rejimidagi tekshiruvlar bilan cheklanmasdan, yadro xotirasi, drayverlar va mustaqil telemetriya manbalarini ham tahlil qilish zarur.

WinDbg kernel debugging va dump-fayllarni tahlil qilish uchun muhim vosita hisoblanadi. .sympath, .reload, lm, !analyze -v, .bugcheck, !process va stek tahlili buyruqlari yadro holatini o‘rganishda foydali. Biroq ushbu vosita natijalari to‘g‘ri symbols, mos dump turi va tahlilchining malakasiga bog‘liq. Xotiraga yozish yoki noma’lum drayverlarni yuklash kabi xavfli amallarni ishlab turgan tizimda bajarish mumkin emas.

Maqolada taklif etilgan metodika dalillarni saqlash, drayverlarni inventarizatsiya qilish, jarayonlar va steklarni tekshirish, kod yaxlitligini baholash hamda Sysmon, Code Integrity, Driver Verifier va HVCI kabi mexanizmlarni birgalikda qo‘llashga asoslanadi. Bunday ko‘p manbali yondashuv bitta indikatorga ortiqcha tayanish xavfini kamaytiradi va xulosalarning ishonchliligini oshiradi.

Kelgusida metodikani real, ammo xavfsiz laboratoriya sharoitida sinash, oldindan tayyorlangan xotira dump-fayllari bo‘yicha indikatorlarni baholash va aniqlash jarayonini qisman avtomatlashtirish maqsadga muvofiq. Bunda asosiy tamoyil — tizim yaxlitligini saqlash, dalillarni takror tekshirish va har bir xulosani ishonchli manba bilan asoslashdir.

Foydalanilgan adabiyotlar

1. MITRE ATT&CK. Rootkit (T1014). Enterprise Techniques. https://attack.mitre.org/techniques/T1014/ (murojaat sanasi: 09.11.2026).
2. Microsoft. Get started with WinDbg (kernel-mode). Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/getting-started-with-windbg--kernel-mode- (murojaat sanasi: 09.11.2026).
3. Microsoft. Microsoft recommended driver block rules. Windows Security documentation. https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/microsoft-recommended-driver-block-rules (murojaat sanasi: 09.11.2026).
4. Microsoft Sysinternals. Sysmon. Event ID 6: Driver Load. https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon (murojaat sanasi: 09.11.2026).
5. Microsoft. Driver security guidance. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/driversecurity/ (murojaat sanasi: 09.11.2026).
6. Microsoft. Driver security checklist. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/driversecurity/driver-security-checklist (murojaat sanasi: 09.11.2026).
7. Microsoft. Analyze a kernel-mode dump file by using WinDbg. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/analyzing-a-kernel-mode-dump-file-with-windbg (murojaat sanasi: 09.11.2026).
8. MITRE ATT&CK. Detection Strategy: Kernel/User-Level Rootkit Behavior Across Platforms (DET0377). https://attack.mitre.org/detectionstrategies/DET0377/ (murojaat sanasi: 09.11.2026).
9. Hoglund G., Butler J. Rootkits: Subverting the Windows Kernel. Boston: Addison-Wesley, 2005.
10. Microsoft. Windows security model for driver developers. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/driversecurity/windows-security-model (murojaat sanasi: 09.11.2026).
11. Yosifovich P., Ionescu A., Russinovich M.E., Solomon D.A. Windows Internals, Part 1: System Architecture, Processes, Threads, Memory Management, and More. 7th ed. Redmond: Microsoft Press, 2017.
12. Ligh M.H., Case A., Levy J., Walters A. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. Indianapolis: Wiley, 2014.
13. Microsoft. Driver Verifier. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/devtest/driver-verifier (murojaat sanasi: 09.11.2026).
14. Microsoft. Implement Memory Integrity compatible code. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/driversecurity/implement-hvci-compatible-code (murojaat sanasi: 09.11.2026).
15. Microsoft. Memory integrity and virtualization-based security. Windows Drivers documentation. https://learn.microsoft.com/en-us/windows-hardware/drivers/bringup/device-guard-and-credential-guard (murojaat sanasi: 09.11.2026).