CERT.LV

Informācijas tehnoloģiju drošības incidentu novēršanas institūcija

Pierādījumu apkopošana pēc kiberincidenta

Incidentu risināšana: #1

Andrejs Konstantinovs, CERT.LV, 26.08.2021

Semināra instruktors

Vārds: Andrejs Konstantinovs

Amats: CERT.LV incidentu risinātājs

Kontakti: kursi@cert.lv, andrejs@cert.lv

Instruktora blogs: caurumi.lv

GPG atslēga: 679A C8D4 A391 6736 D558 07C1 D3D9 0B7C 666A EDCD

Organizatoriska informācija

- Apliecinājumi par dalību nedēļas laikā
- Septembra sākumā tiks izsludināts "Kiberšoks 2021"
  - ar CTF!
- Izvērtējam iespēju rīkot līdzīgus seminārus biežāk
  - būsim priecīgi saņemt jautājumus, komentārus, ieteikumus: kursi@cert.lv

Brīdinājums!

Šajā seminārā apskatīti incidenti, kuros varētu būt nepieciešamība iesaistīt CERT.LV:

- ļaunatūra
- ekspluatētas ievainojamības (parasti publiski pieejamos servisos)
- FIN vai APT grupu operācijas

Cita veida incidentos rīcības plāns var atšķirties. Sevišķi jāpievērš uzmanība tam, vai plānots izmantot savāktos pierādījumus tiesā.

Pierādījumu savākšana tiesai

Ja plānots izmantot savāktos pierādījumus tiesai, tad noteikti jāprecizē plānotās darbības ar savas organizācijas juristiem un / vai Valsts policiju.

Valsts policijā ar kibernoziegumu izmeklēšanu parasti nodarbojas Galvenās kriminālpolicijas pārvaldes Ekonomisko noziegumu apkarošanas pārvaldes 3. nodaļa.

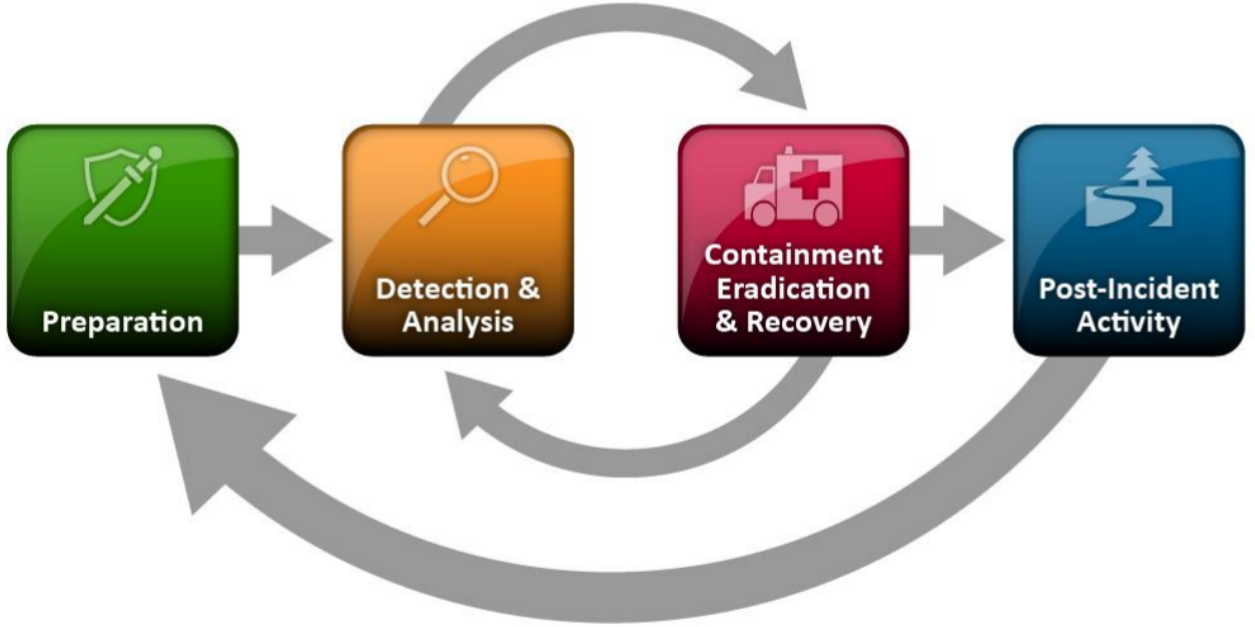
Piezīmes:

- Dažos gadījumos pierādījumu vākšanu labāk atstāt policijai
- Ja iespējams, triāžai izmanto datus, kas jau savākti un atrodas *off-device* (žurnālfaili no centralizētas logošanas sistēmas, rezerves kopijas)
- Vācot pierādījumus patstāvīgi, izvēlas vismazāk invazīvās operācijas (piem. operatīvās atmiņas attēlu un pilnu diska kopiju)
- Pierādījumus savāc liecinieku klātbūtnē, pieraksta hešsummas, oriģinālos pierādījumus ieliek seifā, bet darbojas tikai ar šo pierādījumu kopijām

Semināra plāns

1. Triāžas pamati
2. Pierādījumu vākšanas procedūra
3. Diska attēla un failu vākšana
4. Windows Event Log
5. Operatīvās atmiņas attēla iegūšana

Incidentu risināšana



Computer Security Incident Handling Guide (NIST)
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

Sagatavošanās incidentam

- Svarīgākā fāze
- Kontaktu, komunikāciju protokolu saraksts
- Programmatūra pierādījumu vākšanai un triāžai
- Tīkla dokumentācija
- Lietotās programmatūras saraksts
- *Business Continuity* plāni
- Šis seminārs

Piemērs #1

Pierādījumu savākšanas plāns:

1. formatē ārējo nesēju
2. uzkopē uz ārēja nesēja nepieciešamās programmas un skriptus
3. pievieno nesēju, palaiž programmu (rezultāti saglabāti turpat)
4. pārnes datus uz citu datoru, kur tie tiks analizēti

Pievienojot vienu datu nesēju vairākām sistēmām, riskējam, ka:

- pirmā sistēma var inficēt datu nesējā izvietotās programmas
- otrā sistēmā strādājoša ļaunatūra var savākt pierādījumus no pirmās sistēmas

Piemērs #2

Pierādījumu savākšanas plāns, izmantojot SMB, NFS vai SFTP/FTPS:

1. nepieciešamās programmas un skriptus izvieto uz *read-only* attālinātā diska
2. izveido konkrētai sistēmai paredzētu attālinātu disku (*read-write*)
3. programmas laiž no *read-only* diska, rezultātus ieraksta *read-write* diskā, kas pieejamas tikai šai vienai sistēmai

Triāža

Pirmajā pasaules karā pacienti sadalīti:

1. kas izdzīvos, neatkarībā no saņemtās palīdzības
2. kas neizdzīvos, neatkarībā no saņemtās palīdzības
3. kam steidzami saņemta palīdzība uzlabos izdzīvošanas izredzes

Drošības incidentu triāža

Darbības

- kas jāveic steidzami
- kas ietekmē incidenta gaitu

Drošības incidentu triāža

Incidenta triāžas soļi:

1. potenciāla incidenta atpazīšana
2. verifikācija
3. ietekmes noteikšana
4. incidenta novēršana vai iespaida mazināšana

Potenciāla incidenta atpazīšana

Avotu iedalījums

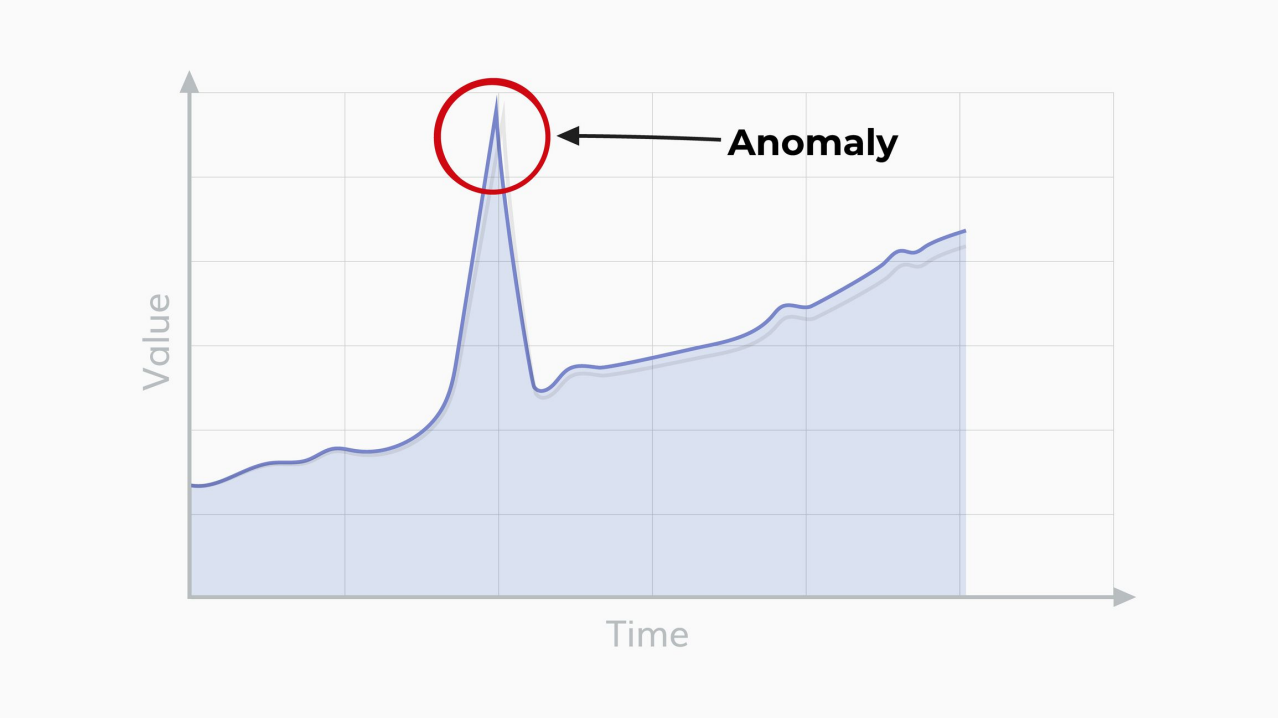
1. ārēji
2. iekšēji

1. pasīvi
2. aktīvi

Potenciāla incidenta atpazīšana

- Tehniskie risinājumi:
  - IDS, SIEM
  - A/V, EDR
  - Antispam, DMARC atskaites
- Cilvēki:
  - Administratori, citi IT darbinieki
  - Klientu atbalsts
  - Pētnieki

Anomālijas



Investopedia

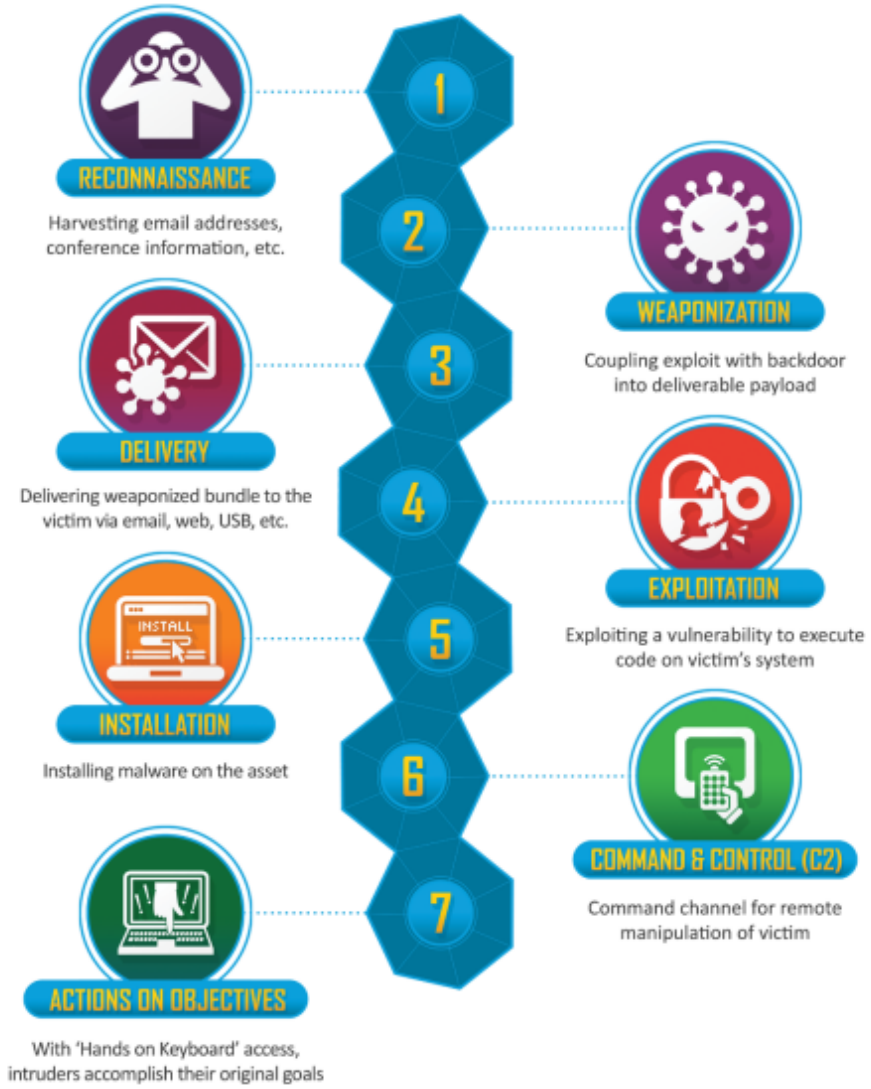
Garbage in, garbage out

- datu analīzes pamatproblēmas:
  - trūkstoši dati
  - lieki dati (*adata siena kaudzē*)
- kā zināt, kas jāvāc?
  - monitoringam
  - pierādījumu saglabāšanai pēc incidenta

Kā zināt, kas jāvāc?

- subjektīva pieredze
- *data-flow* diagrammas
- draudu modelēšana (*Threat Modeling*)
  - OWASP: Threat Modeling Cheat Sheet
  - Threat Modeling: Designing for Security, A. Shostack

formālas diagrammas reti kad eksistē (un ir *up to date*), bet datu plūsma ir noderīgs veids kā konceptuāli skatīties uz incidentu



lockheedmartin.com

Cyber Kill Chain vs MITRE ATT&CK

MITRE ATT&CK™

https://attack.mitre.org/

For real though

- Jebkurš ētiskās hakošanas (*penetration testing*) kurss
  - *adversarial thinking*
  - var aizvietot ar Capture-the-Flag (CTF)
- real-world *Tactics, Techniques, Procedurs (TTPs)* pētīšana
  - uzbrucējiem: https://attack.mitre.org/groups/
  - ļaunatūrai: https://malpedia.caad.fkie.fraunhofer.de/

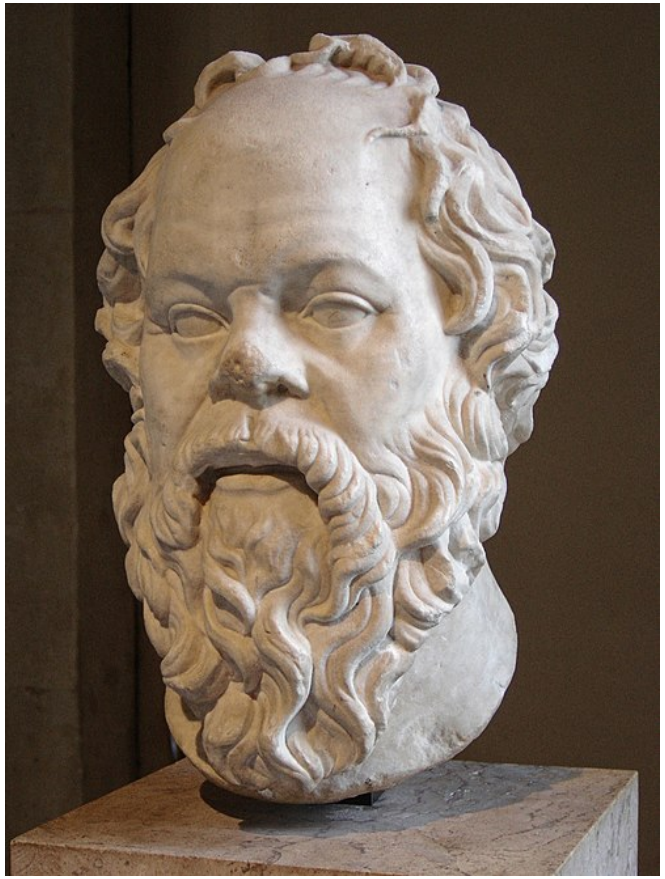
Pierādījumu vākšanas principi

- *most volatile first*:
  - tīkla komunikācija
  - operatīvā atmiņa
  - žurnālfaili
  - diska attēls
- paturam atmiņā šifrēšanu
- mēģinam būt kaķis un nevis pele

+ veikto darbību dokumentācija

Nevar iekāpt divreiz vienā un tai pašā upē.

Sokrats



Ar katru veikto darbību riskējam:
- pārrakstīt / dzēst kādus pierādījumus
- informēt uzbrucēju par izmeklēšanas gaitu

Bet tāpat arī vilcinoties veikt darbības riskējam, ka uzbrucējs:
- turpina datu eksfiltrāciju vai iesāk destruktīvas operācijas
- dzēš pēdas vai padara pierādījumu vākšanu neiespējamu

https://en.wikipedia.org/wiki/Socrates#/media/File:Socrate_du_Louvre.jpg

Pierādījumu kategorijas

- Tīkla komunikācija
- Operatīvā atmiņa
- Faili
  - žurnālfaili, ļaunatūra, reģistrs
  - sistēmas faili var būt aizsargāti un nelasāmi
- Failu sistēmas struktūras
  - NTFS: *Master File Table* ($MFT) + citi faili

Diska attēls *vs* Failu vākšana

- Diska attēls
  - (+) satur visus sistēmas failus
  - (+) žurnālfaili, sistēmas faili
  - (+) arī tas, par ko neesam padomājuši
  - (-) lēns process, ja iegūst no *live* sistēmas
  - (-) samērā liels datu apjoms

Vai ar 100GB ārējo nesēju noteikti pietiks 100GB diska attēla veidošanai?

Diska attēls *vs* Failu vākšana

- Atsevišķu failu vākšana
  - (+) arī no diska attēla būtu jāizņem tie paši faili
  - (+) mazāks datu apjoms
  - (+) var ātrāk uzsākt analīzi
  - (-) varam piemirst kādu failu
  - (-) laikus nepamanītas kļūdas -> trūkstoša informācija

Demo time

- Diska un atsevišķu failu vākšana
  - CyLR: https://github.com/orlikoski/CyLR
  - FTK Imager: https://accessdata.com/product-download/ftk-imager-version-4-5

CyLR

Var izmantot jebkuru failu vākšanai (ieskaitot reģistru, $MFT un citus sistēmas failus). Pēc noklusējuma savāc lielāko daļu nepieciešamā (saraksts norādīts README.md, vēl precīzāk var skatīties CollectionPaths.cs).

Standarta konfigurācijas failu var pārrakstīt (vai papildināt) ar savai videi pielāgotu konfigurāciju.

"CyLR.exe -c overwrite.cfg" ignorē standarta konfigurāciju un savāc tikai to, kas norādīts iekš overwrite.cfg. Savukārt "CyLR.exe -d additional.cfg" savāks gan standarta failus, gan to, kas norādīts iekš additional.cfg.

Konfigurācijas failā var norādīt, gan precīzus failus/direktorijas, gan arī izmantot glob / regex, sk. piemērus.

Windows Event Logs

- Satur informāciju par
  - sistēmas darbību
  - lietotāju darbībām
  - sastaptajām kļūdām
  - atsevišķu servisu žurnālfailus

Kas nav atrodams iekš Event Logs?

- Tas, kas tur netika ierakstīts:
  - Security events
    - 4688: process creation
      - GPO: "Include command line in process creation events"
      - sha hash: Application and Services Logs\Microsoft\Windows\AppLocker
  - PowerShell auditing
    - 4104: script block (first use)
  - Sysmon
    - 11: file creation
    - 23: file delete

Kas nav atrodams iekš Event Logs?

- Tas, kas bija ierakstīts, bet tika dzēsts:
  - tipiski Event Logs apjoms tiek ierobežots pēc to apjoma
  - vecie ieraksti tiek dzēsti, nevis arhivēti
- Risinājumi
  - Computer Configuration → Policies → Administrative Templates → Windows Components → Event Log Service
  - citām kategorijām parametri jāmaina reģistrā
  - centralizēta logošanas sistēma

Ieteicamie auditēšanas iestatījumi

- CERT.LV: https://cert.lv/lv/2020/04/rekomendacijas-auditesanas-iestatijumiem-windows-domena-infrastruktura
- PowerShell auditing (FireEye): https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html
- Sysmon: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Demo time

- Windows Event logs parsēšana
  - PowerShell
  - Eric Zimmerman's tools: https://ericzimmerman.github.io/

Operatīvā atmiņa

Vērtīgs informācijas avots, jo satur:

- izpildītos procesus;
- atvērtos failus, to saturu;
- tīkla komunikācijas artefaktus;
- šifrēšanas atslēgas un atšifrētus datus;
- paroles, lietojumu sesijas, autentifikācijas marķierus (tokens);
- atpakotu programmu kodu;
- ļaunatūras kodu, kas injicēts citās programmās vai bibliotēkās;
- modificētas OS datu struktūras (liecina par rootkit).

Operatīvā atmiņa

Attēlu iegūšanas veidi:

- virtualizācijas hosts
- *live* sistēma
- *crash dumps* ("BSOD")
- "cold boot"

Attēla iegūšana no Hyper-V hosta

1. jāuzinstalē "Debugging Tools for Windows"
2. jāuzinstalē LiveKD no Sysinternals
3. livekd64.exe -y ‘srv*c:\symbols*https://msdl.microsoft.com/download/symbols’ -hv myVM -p -o myVM.dmp -vsym

Pilns Hyper-V virtuālās mašīnas eksports:

Export-VM -Name myVM -Path exported/ -CaptureLiveState CaptureSavedState

Attēla iegūšana no VMWare hosta

Izmantojam iebūvēto *Snapshots* funkcionalitāti.

Nepieciešami divi faili:

1. .vmss
2. .mem

Attēla iegūšana no VirtualBox hosta

Iebūvētā *Snapshots* funkcionalitāte neder, jāizmanto komandrinda:

VBoxManage debugvm "myVM" dumpvmcore --filename=myVM.dmp

Attēla iegūšana no libvirt (KVM/QEMU) hosta

Jāizmanto komandrinda:

virsh -c qemu:///system dump --memory-only myVM myVM.dmp

Jaunu Volatility profilu veidošana

Nepieciešama, ja Windows versija netiek atbalstīta.

Gadās reti, bet drošības pēc savācot atmiņas attēlu, vēlams saglabāt arī:

c:\windows\system32\ntoskrnl.exe

Vairāk informācijas: https://www.osdfcon.org/presentations/2020/Jamie-Levy_Troubleshooting-Memory.pdf

Demo time

Attēlu iegūšana ar:

- WinPMEM: https://github.com/Velocidex/WinPmem
- FTK Imager: https://accessdata.com/product-download/ftk-imager-version-4-5

Materiāla pārbaude ar:

- Volatility (2, 3)

Vai uzbrucējiem šī informācija ir noderīga?

https://cert.lv/lv/2021/03/it-drosibas-incidenta-pieradijumu-materiala-iegusana-operativa-atmina

WinPMEM

Parasti vienkārši palaižam *standalone* programmu, norādot atmiņas attēla ceļu:

winpmem_mini_x64_rc2.exe memory.dmp

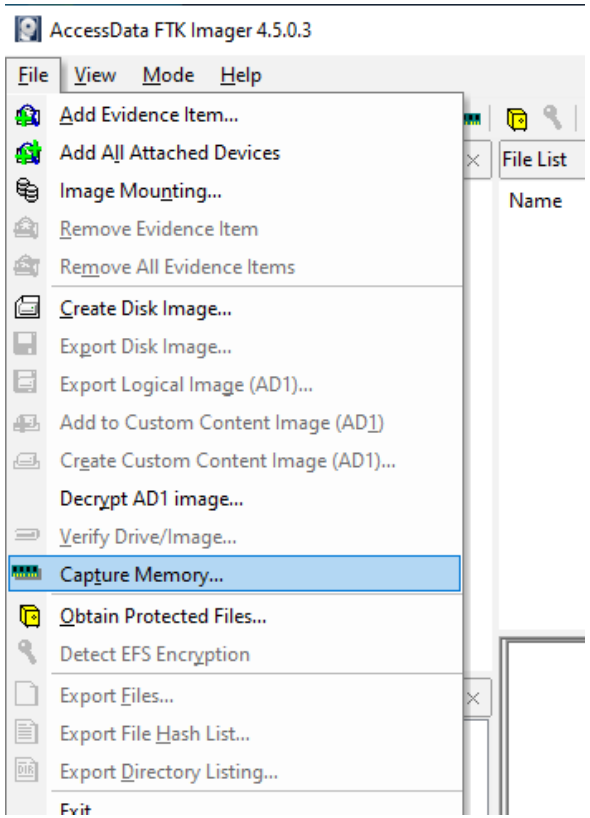
Dažreiz var būt nepieciešams izmantot citas metodes (pēc noklusējuma uz AMD64 tiek izmantots *PTE Remapping* un parasti tā arī ir viskorektākā):

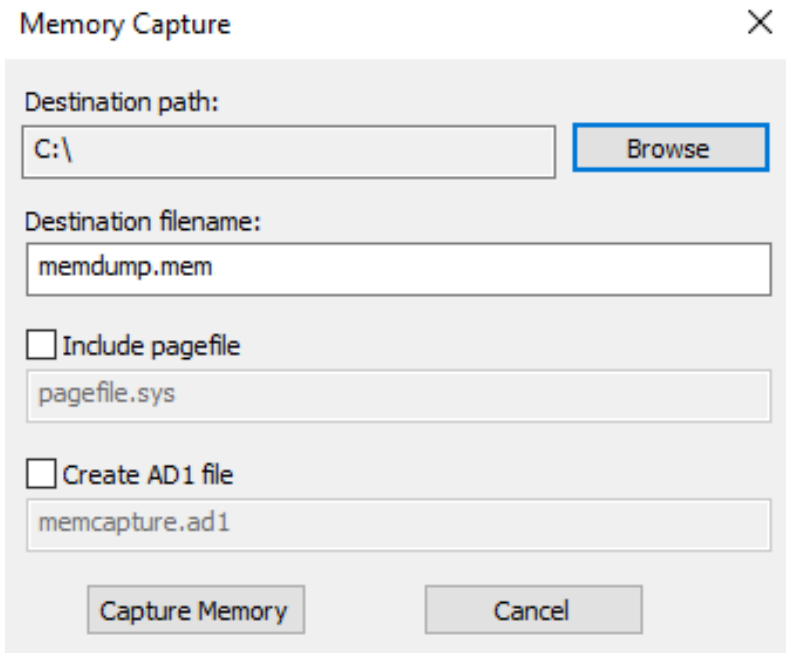
winpmem_mini_x64_rc2.exe -0 method_0.dmp

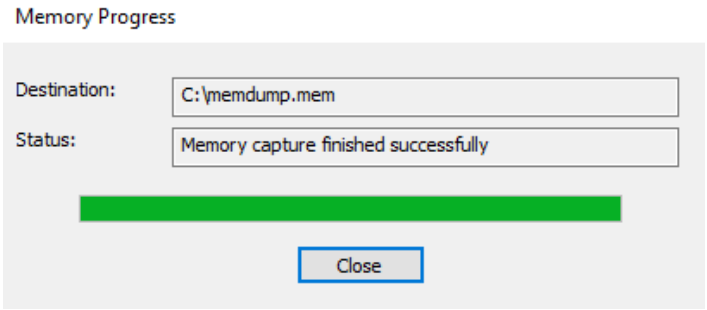
winpmem_mini_x64_rc2.exe -1 method_1.dmp

Jaunajās Windows versijās -0 tiek bloķēts un atmiņas attēls tiek iegūts nekorekts (var pārbaudīt ar Volatility, kā aprakstīts tālākos).

FTK Imager


AccessData FTK Imager 4.5.0.3
File View Mode Help
Add Evidence Item...
Add All Attached Devices
Image Mounting...
Remove Evidence Item
Remove All Evidence Items
Create Disk Image...
Export Disk Image...
Export Logical Image (AD1)...
Add to Custom Content Image (AD1)
Create Custom Content Image (AD1)...
Decrypt AD1 image...
Verify Drive/Image...
Capture Memory...
Obtain Protected Files...
Detect EFS Encryption
Export Files...
Export File Hash List...
Export Directory Listing...
Exit
File List
Name


Memory Capture
Destination path:
C:\
Browse
Destination filename:
memdump.mem
Include pagefile
pagefile.sys
Create AD1 file
memcapture.ad1
Capture Memory
Cancel


Memory Progress
Destination:
C:\memdump.mem
Status:
Memory capture finished successfully
Close

Attēla pārbaude ar Volatility

Pārbaudes mērķis - pārliecināties, ka savāktais atmiņas attēls iegūts korekti (bez *page smear*) un ir lasāms.

Volatility ir pamatrīks, kas tiek izmantots operatīvās atmiņas analīzei. Tam ir divas versijas:

1. Šo darbību noteikti nedrīst veikt uz datora, no kura tiek vākti pierādījumi.
2. Vēlams izmantot atsevišķu Linux virtuālo mašīnu, kas tiek atjaunota uz sākotnējo snapshot pēc pārbaudes.

- Volatility 2 (legacy)
  - izmanto Python 2
  - ir standalone .exe, bet sen nebija atjaunots, neatbalsta jaunās Win10 un Win2019 versijas, ieteicams izmantot git versiju
  - manuāli jānorāda profils, kas atbilst Windows versijai
- Volatility 3 (ieteicams)
  - izmanto Python 3
  - nav jānorāda profils, Windows versija tiek noteikta automātiski

Volatility 3

Nepieciešams Python 3 un git. Var izmantot Linux virtuālo mašīnu, Windows Subsystem for Linux vai chocolatey:

```
choco install python3 git
git clone https://github.com/volatilityfoundation/volatility3
```

Atmiņas attēlu var pārbaudīt ar komandu:

```
python volatility3\vol.py -f memory.dmp windows.pstree
```

Volatility 2

Novecojusi versija, kas netiek attīstīta, bet internetā par to atrodams vairāk informācijas kā par Volatility 3. Tā joprojām darbojas, ja nepieciešams analizēt vecākas Windows versijas (7, 8, 2012, 2016). Jaunākām Windows 10/11 un Windows Server 2019 versijām parasti jāveido profili patstāvīgi, turklāt tiek ieviests aizvien vairāk jaunu drošības funkciju, kas nav savietojamas ar Volatility 2 - **šajos gadījumos iesakāms izmantot Volatility 3**.

Vēlams izmantot Linux VM, kur jāpieinstalē Python 2 un nepieciešamie moduļi:

```
sudo apt install python2 python2-dev
virtualenv venv && . venv/bin/activate
pip install pycrypto distorm3
git clone https://github.com/volatilityfoundation/volatility
python2 volatility/vol.py --profile Win2012R2x64_18340 -f memory.dmp
```

Volatility 2 profili

Volatility 2 profili apraksta Windows datu struktūru atrašanās vietu atmiņas attēlā. Pilnu profilu sarakstu lietotai Volatility 2 versijai var iegūt ar, palaižot

```
python volatility/vol.py --info
```

Profilam jābūt gana tuvam, bet nav precīzi jāsakrīt ar Windows versiju.

```
Win10x64              - A Profile for Windows 10 x64
Win10x64_10240_17770  - A Profile for Windows 10 x64 (10.0.10240.17770 / 2018-02-10)
Win10x64_17134        - A Profile for Windows 10 x64 (10.0.17134.1 / 2018-04-11)
Win10x64_17763        - A Profile for Windows 10 x64 (10.0.17763.0 / 2018-10-12)
Win10x64_18362        - A Profile for Windows 10 x64 (10.0.18362.0 / 2019-04-23)
Win10x64_19041        - A Profile for Windows 10 x64 (10.0.19041.0 / 2020-04-17)
Win2012R2x64          - A Profile for Windows Server 2012 R2 x64
Win2012R2x64_18340    - A Profile for Windows Server 2012 R2 x64 (6.3.9600.18340 / 2016-05-13)
Win2012x64            - A Profile for Windows Server 2012 x64
Win2016x64_14393      - A Profile for Windows Server 2016 x64 (10.0.14393.0 / 2016-07-16)
```

Jaunus profilus var izveidot, izmantojot Windows kodolu, kas atrodams failā c:\windows\system32\ntoskrnl.exe, tāpēc vēlams vienmēr saglabāt šo failu, veidojot atmiņas attēlu.

Vairāk informācijas: https://www.osdfcon.org/presentations/2020/Jamie-Levy_Troubleshooting-Memory.pdf

Atmiņas attēla pārbaude ar Volatility 2

Attēlu var pārbaudīt, izmantojot Volatility 2 komandu *imageinfo*:

python volatility/vol.py --profile Win2016x64_14393 **-f memory.dmp imageinfo**

Var laist arī citas Volatility 2 komandas, piemēram pārbaudīt procesu sarakstu ar:

python volatility/vol.py --profile Win2016x64_14393 **-f memory.dmp pstree**

Paldies par uzmanību!

- Apliecinājumi par dalību nedēļas laikā
- Septembra sākumā tiks izsludināts "Kiberšoks 2021"
  - ar CTF!
- Izvērtējam iespēju rīkot līdzīgus seminārus biežāk
  - būsim priecīgi saņemt jautājumus, komentārus, ieteikumus: kursi@cert.lv