Deobfuscate/Decode Files or Information, Technique T1140 - Enterprise

Archived: 2026-04-05 17:46:08 UTC

S0469 ABK

ABK has the ability to decrypt AES encrypted payloads.[5]

S1028 Action RAT

Action RAT can use Base64 to decode actor-controlled C2 server communications.[6]

S0331 Agent Tesla

Agent Tesla has the ability to decrypt strings encrypted with the Rijndael symmetric encryption algorithm.[7]

G1030 Agrius

Agrius has deployed base64-encoded variants of ASPXSpy to evade detection.[8]

S1025 Amadey

Amadey has decoded antivirus name strings.[9]

S1133 Apostle

Apostle compiled code is obfuscated in an unspecified fashion prior to delivery to victims.[8]

S0584 AppleJeus

AppleJeus has decoded files received from a C2.[10]

S0622 AppleSeed

AppleSeed can decode its payload prior to execution.[11]

G0073 APT19

An APT19 HTTP malware variant decrypts strings using single-byte XOR keys.[12]

G0007 APT28

An APT28 macro uses the command `certutil -decode` to decode contents of a .txt file storing the base64 encoded payload.[13][14]

C0051 APT28 Nearest Neighbor Campaign

During APT28 Nearest Neighbor Campaign, APT28 unarchived data using the GUI version of WinRAR.[15]

G0082 APT38

APT38 has used the RC4 algorithm to decrypt configuration data. [16]

G0087 APT39

APT39 has used malware to decrypt encrypted CAB files.[17]

C0046 ArcaneDoor

ArcaneDoor involved the use of Base64 obfuscated scripts and commands.[18]

S0456 Aria-body

Aria-body has the ability to decrypt the loader configuration and payload DLL.[19]

S0373 Astaroth

Astaroth uses a fromCharCode() deobfuscation method to avoid explicitly writing execution commands and to hide its code. [20][21]

S0347 AuditCred

AuditCred uses XOR and RC4 to perform decryption on the code functions.[22]

S0640 Avaddon

Avaddon has decrypted encrypted strings.[23]

S0473 Avenger

Avenger has the ability to decrypt files downloaded from C2.[5]

S1053 AvosLocker

AvosLocker has deobfuscated XOR-encoded strings.[24]

S0344 Azorult

Azorult uses an XOR key to decrypt content and uses Base64 to decode the C2 address.[25][26]

S0638 Babuk

Babuk has the ability to unpack itself into memory using XOR.[27][28]

S0414 BabyShark

BabyShark has the ability to decode downloaded files prior to execution.[29]

S0475 BackConfig

BackConfig has used a custom routine to decrypt strings.[30]

S0642 BADFLICK

BADFLICK can decode shellcode using a custom rotating XOR cipher.[31]

S0234 Bandook

Bandook has decoded its PowerShell script.[32]

S0239 Bankshot

Bankshot decodes embedded XOR strings.[33]

S0534 Bazar

Bazar can decrypt downloaded payloads. Bazar also resolves strings and other artifacts at runtime.[34][35]

S0470 BBK

BBK has the ability to decrypt AES encrypted payloads.[5]

S0127 BBSRAT

BBSRAT uses Expand to decompress a CAB file into executable content.[36]

S0574 BendyBear

BendyBear has decrypted function blocks using a XOR key during runtime to evade detection.[37]

S0268 Bisonal

Bisonal has decoded strings in the malware using XOR and RC4.[38][39]

G1043 BlackByte

BlackByte has encoded commands in base64-encoded sections concatenated together in PowerShell.[40]
BlackByte uses PowerShell commands to disable Windows Defender.[41]

S1180 BlackByte Ransomware

BlackByte Ransomware is distributed as an obfuscated JavaScript launcher file.[42]

S0520 BLINDINGCAN

BLINDINGCAN has used AES and XOR to decrypt its DLLs.[43]

S1226 BOOKWORM

BOOKWORM has decoded its Base64 encoded payload prior to execution.[44] BOOKWORM has also encrypted files with RC4 and has decrypted its payload prior to execution.[45]

S0635 BoomBox

BoomBox can decrypt AES-encrypted files downloaded from C2.[46]

S0415 BOOSTWRITE

BOOSTWRITE has used a a 32-byte long multi-XOR key to decode data inside its payload.[47]

G0060 BRONZE BUTLER

BRONZE BUTLER downloads encoded payloads and decodes them on the victim.[48]

S1063 Brute Ratel C4

Brute Ratel C4 has the ability to deobfuscate its payload prior to execution.[49]

S1039 Bumblebee

Bumblebee can deobfuscate C2 server responses and unpack its code on targeted hosts.[50][51]

S0482 Bundlore

Bundlore has used `openssl` to decrypt AES encrypted payload data. Bundlore has also used base64 and RC4 with a hardcoded key to deobfuscate data.[52]

S1118 BUSHWALK

BUSHWALK can Base64 decode and RC4 decrypt malicious payloads sent through a web request's command parameter.[53][54]

C0017 C0017

During C0017, APT41 used the DUSTPAN loader to decrypt embedded payloads.[55]

C0021 C0021

During C0021, the threat actors deobfuscated encoded PowerShell commands including use of the specific string `'FromBase'+0x40+'String'`, in place of `FromBase64String` which is normally used to decode base64.[56][57]

S0335 Carbon

Carbon decrypts task and configuration files for execution.[58][59]

S0348 Cardinal RAT

Cardinal RAT decodes many of its artifacts and is decrypted (AES-128) after being downloaded.[60]

S1224 CASTLETAP

CASTLETAP can filter and deobfuscate an XOR encrypted activation string in the payload of an ICMP echo request.[61]

S0160 certutil

certutil has been used to decode binaries hidden inside certificate files as Base64 information.[1]

S0631 Chaes

Chaes has decrypted an AES encrypted binary file to trigger the download of other files.[62]

S0674 CharmPower

CharmPower can decrypt downloaded modules prior to execution.[63]

S1149 CHIMNEYSWEEP

CHIMNEYSWEEP can use an embedded RC4 key to decrypt Windows API function strings.[64]

S1041 Chinoxy

The Chinoxy dropping function can initiate decryption of its config file.[65]

S0667 Chrommme

Chrommme can decrypt its encrypted internal code.[66]

G1021 Cinnamon Tempest

Cinnamon Tempest has used weaponized DLLs to load and decrypt payloads.[67]

S1236 CLAIMLOADER

CLAIMLOADER has decoded its payload prior to execution.[68][69]

S0660 Clambling

Clambling can deobfuscate its payload prior to execution.[70][71]

S0611 Clop

Clop has used a simple XOR operation to decrypt strings.[72]

S1105 COATHANGER

COATHANGER decodes configuration items from a bundled file for command and control activity.[73]

S0154 Cobalt Strike

Cobalt Strike can deobfuscate shellcode using a rolling XOR and decrypt metadata from Beacon sessions.[74][75]

S0369 CoinTicker

CoinTicker decodes the initially-downloaded hidden encoded file using OpenSSL.[76]

S0126 ComRAT

ComRAT has used unique per machine passwords to decrypt the orchestrator payload and a hardcoded XOR key to decrypt its communications module. ComRAT has also used a unique password to decrypt the file used for its hidden file system.[77][78]

S0575 Conti

Conti has decrypted its payload using a hardcoded AES-256 key.[79][80]

S0492 CookieMiner

CookieMiner has used Google Chrome's decryption and extraction operations.[81]

S1235 CorKLOG

CorKLOG has decoded XOR encrypted strings.[82]

S0614 CostaBricks

CostaBricks has the ability to use bytecode to decrypt embedded payloads.[83]

S0115 Crimson

Crimson can decode its encoded PE file prior to execution.[84]

S1153 Cuckoo Stealer

Cuckoo Stealer strings are deobfuscated prior to execution.[85][86]

S0687 Cyclops Blink

Cyclops Blink can decrypt and parse instructions sent from C2.[87]

S1014 DanBot

DanBot can use a VBA macro to decode its payload prior to installation and execution.[88]

S1111 DarkGate

DarkGate installation includes binary code stored in a file located in a hidden directory, such as `shell.txt`, that is decrypted then executed.[89] DarkGate uses hexadecimal-encoded shellcode payloads during installation that are called via Windows API `CallWindowProc()` to decode and then execute.[90]

G0012 Darkhotel

Darkhotel has decrypted strings and imports using RC4 during execution.[91][92]

S1066 DarkTortilla

DarkTortilla can decrypt its payload and associated configuration elements using the Rijndael cipher.[93]

S0673 DarkWatchman

DarkWatchman has the ability to self-extract as a RAR archive.[94]

S0255 DDKONG

DDKONG decodes an embedded configuration using XOR.[95]

S1052 DEADEYE

DEADEYE has the ability to combine multiple sections of a binary which were broken up to evade detection into a single .dll prior to execution.[55]

S1134 DEADWOOD

DEADWOOD XORs some strings within the binary using the value `0xD5`, and deobfuscates these items at runtime.[8]

S0354 Denis

Denis will decrypt important strings used for C&C communication.[96]

S0547 DropBook

DropBook can unarchive data downloaded from the C2 to obtain the payload and persistence modules.[97]

S0502 Drovorub

Drovorub has de-obsfuscated XOR encrypted payloads in WebSocket messages.[98]

S0567 Dtrack

Dtrack has used a decryption routine that is part of an executable physical patch.[99]

S1158 DUSTPAN

DUSTPAN decodes and decrypts embedded payloads.[100]

S1159 DUSTTRAP

DUSTTRAP deobfuscates embedded payloads.[100]

S0024 Dyre

Dyre decrypts resources needed for targeting the victim.[101][102]

G1006 Earth Lusca

Earth Lusca has used certutil to decode a string into a cabinet file.[103]

S0377 Ebury

Ebury has verified C2 domain ownership by decrypting the TXT record using an embedded RSA public key.[104]

S0624 Ecipekac

Ecipekac has the ability to decrypt fileless loader modules.[105]

S0554 Egregor

Egregor has been decrypted before execution.[106][107]

S1247 Embargo

Embargo has utilized MDeployer to decrypt two payloads that contain MS4Killer toolkit b.cache and the Embargo ransomware executable a.cache with a hardcoded RC4 key `wlQYLoPCil3niI7x8CvR9EtNtL/aeaHrZ23LP3fAsJogVTIzdnZ5Pi09ZVeHFkiB`.[108]

S0367 Emotet

Emotet has used a self-extracting RAR file to deliver modules to victims. Emotet has also extracted embedded executables from files using hard-coded buffer offsets.[109]

S0634 EnvyScout

EnvyScout can deobfuscate and write malicious ISO files to disk.[46]

S0401 Exaramel for Linux

Exaramel for Linux can decrypt its configuration file.[110]

S1179 Exbyte

Exbyte decodes and decrypts data stored in the configuration file with a key provided on the command line during execution.[111]

S0361 Expand

Expand can be used to decompress a local or remote CAB file into an executable.[112]

S0512 FatDuke

FatDuke can decrypt AES encrypted C2 communications.[113]

G1016 FIN13

FIN13 has utilized `certutil` to decode base64 encoded versions of custom malware.[114]

G0046 FIN7

FIN7 has decoded a malicious PowerShell script using `certutil -decode hex` and has decoded an XOR-obfuscated block of data with the key `qawsed1q2w3e`, which led to the installation of Lizar.[115]

S0355 Final1stspy

Final1stspy uses Python code to deobfuscate base64-encoded strings.[116]

S0182 FinFisher

FinFisher extracts and decrypts stage 3 malware, which is stored in encrypted resources.[117][118]

S0618 FIVEHANDS

FIVEHANDS has the ability to decrypt its payload prior to execution.[119][120][121]

S0661 FoggyWeb

FoggyWeb can be decrypted in memory using a Lightweight Encryption Algorithm (LEA)-128 key and decoded using a XOR key.[122]

S1120 FRAMESTING

FRAMESTING can decompress data received within `POST` requests.[53]

C0001 Frankenstein

During Frankenstein, the threat actors deobfuscated Base64-encoded commands following the execution of a malicious script, which revealed a small script designed to obtain an additional payload.[123]

S0628 FYAnti

FYAnti has the ability to decrypt an embedded .NET module.[105]

G0047 Gamaredon Group

Gamaredon Group tools decrypted additional payloads from the C2. Gamaredon Group has also decoded Base64-encoded source code of a downloader.[124][125][126] Additionally, Gamaredon Group has decoded Telegram content to reveal the IP address for C2 communications.[127]

S0666 Gelsemium

Gelsemium can decompress and decrypt DLLs and shellcode.[66]

S0032 gh0st RAT

gh0st RAT has decrypted and loaded the gh0st RAT DLL into memory, once the initial dropper executable is launched.[128]

S1117 GLASSTOKEN

GLASSTOKEN has the ability to decode hexadecimal and Base64 C2 requests.[129]

S0588 GoldMax

GoldMax has decoded and decrypted the configuration file when executed.[130][131]

S0477 Goopy

Goopy has used a polymorphic decryptor to decrypt itself at runtime.[96]

S1138 Gootloader

Gootloader has the ability to decode and decrypt malicious payloads prior to execution.[132][133]

G0078 Gorgon Group

Gorgon Group malware can decode contents from a payload that was Base64 encoded and write the contents to a file.[134]

S0531 Grandoreiro

Grandoreiro can decrypt its encrypted internal strings.[135]

S0690 Green Lambert

Green Lambert can use multiple custom routines to decrypt strings prior to execution.[136][137]

S0632 GrimAgent

GrimAgent can use a decryption algorithm for strings based on Rotate on Right (RoR) and Rotate on Left (RoL) functionality.[138]

S0499 Hancitor

Hancitor has decoded Base64 encoded URLs to insert a recipient’s name into the filename of the Word document. Hancitor has also extracted executables from ZIP files.[139][140]

S0697 HermeticWiper

HermeticWiper can decompress and copy driver files using `LZCopy`.[141]

S1249 HexEval Loader

HexEval Loader has decoded its payload prior to execution.[142][143][144]

S1027 Heyoka Backdoor

Heyoka Backdoor can decrypt its payload prior to execution.[145]

S0394 HiddenWasp

HiddenWasp uses a cipher to implement a decoding function.[146]

G0126 Higaisa

Higaisa used certutil to decode Base64 binaries at runtime and a 16-byte XOR key to decrypt data.[147][148]

S0601 Hildegard

Hildegard has decrypted ELF files with AES.[149]

S1097 HUI Loader

HUI Loader can decrypt and load files containing malicious payloads.[150]

S0398 HyperBro

HyperBro can unpack and decrypt its payload prior to execution.[70][151]

S1022 IceApple

IceApple can use a Base64-encoded AES key to decrypt tasking.[152]

S0434 Imminent Monitor

Imminent Monitor has decoded malware components that are then dropped to the system.[153]

S1139 INC Ransomware

INC Ransomware can run `CryptStringToBinaryA` to decrypt base64 content containing its ransom note.[154]

S0604 Industroyer

Industroyer decrypts code to connect to a remote C2 server.[155]

S1245 InvisibleFerret

InvisibleFerret has decoded XOR-encrypted and Base-64-encoded payloads prior to execution.[156]

S0260 InvisiMole

InvisiMole can decrypt, unpack and load a DLL from its resources, or from blobs encrypted with Data Protection API, two-key triple DES, and variations of the XOR cipher.[157][158]

S0581 IronNetInjector

IronNetInjector has the ability to decrypt embedded .NET and PE payloads.[159]

S0189 ISMInjector

ISMInjector uses the `certutil` command to decode a payload file.[160]

C0044 Juicy Mix

During Juicy Mix, OilRig used a script to concatenate and deobfuscate encoded strings in Mango.[161]

S1190 Kapeka

Kapeka utilizes obfuscated JSON structures for various data storage and configuration management items.[162]

G0004 Ke3chang

Ke3chang has deobfuscated Base64-encoded shellcode strings prior to loading them.[163]

S0585 Kerrdown

Kerrdown can decode, decrypt, and decompress multiple layers of shellcode.[164]

S0487 Kessel

Kessel has decrypted the binary's configuration once the `main` function was launched.[165]

S1051 KEYPLUG

KEYPLUG can decode its configuration file to determine C2 protocols.[55]

S0526 KGH_SPY

KGH_SPY can decrypt encrypted strings and write them to a newly created folder.[166]

G0094 Kimsuky

Kimsuky has decoded malicious VBScripts using Base64.[167] Kimsuky has also decoded malicious PowerShell scripts using Base64.[168]

S0641 Kobalos

Kobalos decrypts strings right after the initial communication, but before the authentication process.[169]

S0669 KOCTOPUS

KOCTOPUS has deobfuscated itself before executing its commands.[170]

S0356 KONNI

KONNI has used certutil to download and decode base64 encoded strings and has also devoted a custom section to performing all the components of the deobfuscation process.[171][172]

S0236 Kwampirs

Kwampirs decrypts and extracts a copy of its main DLL payload when executing.[173]

S1160 Latrodectus

Latrodectus has the ability to deobfuscate encrypted strings.[174][175][176]

G0032 Lazarus Group

Lazarus Group has used shellcode within macros to decrypt and manually map DLLs and shellcode into memory at runtime.[177][178]

G0065 Leviathan

Leviathan has used a DLL known as SeDll to decrypt and execute other JavaScript backdoors.[179]

S0395 LightNeuron

LightNeuron has used AES and XOR to decrypt configuration files and commands.[180]

S1119 LIGHTWIRE

LIGHTWIRE can RC4 decrypt and Base64 decode C2 commands.[53]

S1186 Line Dancer

Line Dancer shellcode payloads are base64 encoded when transmitted to compromised devices.[181]

S0513 LiteDuke

LiteDuke has the ability to decrypt and decode multiple layers of obfuscation.[113]

S0681 Lizar

Lizar has decrypted its configuration data, such as the C2 IP address, ports and other network communication.[182][183]

S1199 LockBit 2.0

LockBit 2.0 can decode scripts and strings in loaded modules.[184][185]

S1202 LockBit 3.0

The LockBit 3.0 payload is decrypted at runtime.[186][187][188]

S0447 Lokibot

Lokibot has decoded and decrypted its stages multiple times using hard-coded keys to deliver the final payload, and has decoded its server response hex string using XOR.[189]

S0582 LookBack

LookBack has a function that decrypts malicious data.[190]

S0532 Lucifer

Lucifer can decrypt its C2 address upon execution.[191]

S1213 Lumma Stealer

Lumma Stealer has used Base64-encoded content during execution, decoded via PowerShell.[192]

S1143 LunarLoader

LunarLoader can deobfuscate files containing the next stages in the infection chain.[193]

S1142 LunarMail

LunarMail can decrypt strings to retrieve configuration settings.[193]

S1141 LunarWeb

LunarWeb can decrypt strings related to communication configuration using RC4 with a static key.[193]

S0409 Machete

Machete's downloaded data is decrypted using AES.[194]

S1016 MacMa

MacMa decrypts a downloaded file using AES-128-EBC with a custom delta.[195]

S1060 Mafalda

Mafalda can decrypt files and data.[196]

S1182 MagicRAT

MagicRAT stores command and control URLs using base64 encoding in the malware's configuration file.[197]

G1026 Malteiro

Malteiro has the ability to deobfuscate downloaded files prior to execution.[198]

S1244 Medusa Ransomware

Medusa Ransomware has decoded XOR encrypted strings prior to execution in memory.[199][200]

S0576 MegaCortex

MegaCortex has used a Base64 key to decode its components.[201]

G0045 menuPass

menuPass has used certutil in a macro to decode base64-encoded content contained in a dropper document attached to an email. The group has also used `certutil -decode` to decode files on the victim's machine when dropping UPPERCUT.[202][203]

S0443 MESSAGETAP

After checking for the existence of two files, keyword_parm.txt and parm.txt, MESSAGETAP XOR decodes and read the contents of the files. [204]

S1059 metaMain

metaMain can decrypt and load other modules.[196]

S0455 Metamorfo

Upon execution, Metamorfo has unzipped itself after being downloaded to the system and has performed string decryption.[205][206][207]

S0280 MirageFox

MirageFox has a function for decrypting data containing C2 configuration information.[208]

S1122 Mispadu

Mispadu decrypts its encrypted configuration files prior to execution.[198][209]

G0021 Molerats

Molerats decompresses ZIP files once on the victim machine.[210]

S1026 Mongall

Mongall has the ability to decrypt its payload prior to execution.[145]

G1036 Moonstone Sleet

Moonstone Sleet delivered payloads using multiple rounds of obfuscation and encoding to evade defenses and analysis.[211]

S1221 MOPSLED

MOPSLED can decrypt obfuscated configuration files.[212]

S0284 More_eggs

More_eggs will decode malware components that are then dropped to the system.[213]

S1047 Mori

Mori can resolve networking APIs from strings that are ADD-encrypted.[214]

G0069 MuddyWater

MuddyWater has decoded base64-encoded PowerShell, JavaScript, and VBScript.[215][216][217][218]

G0129 Mustang Panda

Mustang Panda has the ability to decrypt its payload prior to execution.[219][220][44][221] Mustang Panda has also utilized RC4 encryption for malicious payloads.[222][45]

S0637 NativeZone

NativeZone can decrypt and decode embedded Cobalt Strike beacon stage shellcode.[46]

S0457 Netwalker

Netwalker's PowerShell script can decode and decrypt multiple layers of obfuscation, leading to the Netwalker DLL being loaded into memory.[223]

S1147 Nightdoor

Nightdoor stores network configuration data in a file XOR encoded with the key value of `0x7A`.[224]

S1100 Ninja

The Ninja loader component can decrypt and decompress the payload.[225][226]

S0353 NOKKI

NOKKI uses a unique, custom de-obfuscation technique.[227]

S1170 ODAgent

ODAgent can Base64-decode and XOR decrypt received C2 commands.[228]

S1172 OilBooster

OilBooster can Base64-decode and XOR-decrypt C2 commands taken from JSON files.[228]

G0049 OilRig

A OilRig macro has run a PowerShell command to decode file contents. OilRig has also used certutil to decode base64-encoded files on victims.[229][160][230][231]

S0439 Okrum

Okrum's loader can decrypt the backdoor code, embedded within the loader or within a legitimate PNG file. A custom XOR cipher or RC4 is used for decryption.[232]

S0052 OnionDuke

OnionDuke can use a custom decryption algorithm to decrypt strings.[113]

S0264 OopsIE

OopsIE concatenates then decompresses multiple resources to load an embedded .Net Framework assembly.[230]

C0016 Operation Dust Storm

During Operation Dust Storm, attackers used VBS code to decode payloads.[233]

C0006 Operation Honeybee

During Operation Honeybee, malicious files were decoded prior to execution.[234]

C0005 Operation Spalax

For Operation Spalax, the threat actors used a variety of packers and droppers to decrypt malicious payloads.[235]

S0402 OSX/Shlayer

OSX/Shlayer can base64-decode and AES-decrypt downloaded payloads.[236] Versions of OSX/Shlayer pass encrypted and password-protected code to `openssl` and then write the payload to the `/tmp` folder.[237][238]

S0352 OSX_OCEANLOTUS.D

OSX_OCEANLOTUS.D uses a decode routine combining bit shifting and XOR operations with a variable key that depends on the length of the string that was encoded. If the computation for the variable XOR key turns out to be 0, the default XOR key of 0x1B is used. This routine is also referenced as the `rotate` function in reporting.[239]

S0598 P.A.S. Webshell

P.A.S. Webshell can use a decryption mechanism to process a user supplied password and allow execution.[110]

S1050 PcShare

PcShare has decrypted its strings by applying a XOR operation and a decompression using a custom implemented LZM algorithm.[65]

S1145 Pikabot

Pikabot decrypts command and control URIs using ADVobfuscator, and decrypts IP addresses and port numbers with a custom algorithm.[240] Other versions of Pikabot decode chunks of stored stage 2 payload content in the initial payload `.text` section before consolidating them for further execution.[241] Overall LunarMail is associated with multiple encoding and encryption mechanisms to obfuscate the malware's presence and avoid analysis or detection.[242]

S0517 Pillowmint

Pillowmint has been decompressed by included shellcode prior to being launched.[243]

S1031 PingPull

PingPull can decrypt received data from its C2 server by using AES.[244]

S0501 PipeMon

PipeMon can decrypt password-protected executables.[245]

S1123 PITSTOP

PITSTOP can deobfuscate base64 encoded and AES encrypted commands.[54]

S0013 PlugX

PlugX decompresses and decrypts itself using the Microsoft API call RtlDecompressBuffer.[246][70][247] PlugX has also decrypted its payloads in memory.[248][249][220][221]

S0428 PoetRAT

PoetRAT has used LZMA and base64 libraries to decode obfuscated scripts.[250]

S0518 PolyglotDuke

PolyglotDuke can use a custom algorithm to decrypt strings used by the malware.[113]

S1173 PowerExchange

PowerExchange can decode and decrypt C2 commands received via email.[251]

S1012 PowerLess

PowerLess can use base64 and AES ECB decryption prior to execution of downloaded modules.[252]

S0223 POWERSTATS

POWERSTATS can deobfuscate the main backdoor code.[217]

S1046 PowGoop

PowGoop can decrypt PowerShell scripts for execution.[214][253]

S0279 Proton

Proton uses an encrypted file to store commands and configuration values.[254]

S0613 PS1

PS1 can use an XOR key to decrypt a PowerShell loader and payload binary.[83]

S0147 Pteranodon

Pteranodon can decrypt encrypted data strings prior to using them.[255]

S1228 PUBLOAD

PUBLOAD has decoded its payload prior to execution.[249][219][69][256][44]

S0196 PUNCHBUGGY

PUNCHBUGGY has used PowerShell to decode base64-encoded assembly.[257]

S1032 PyDCrypt

PyDCrypt has decrypted and dropped the DCSrv payload to disk.[258]

S0650 QakBot

QakBot can deobfuscate and re-assemble code strings for execution.[259][260][261]

S0269 QUADAGENT

QUADAGENT uses AES and a preshared key to decrypt the custom Base64 routine used to encode strings and scripts.[262]

S1076 QUIETCANARY

QUIETCANARY can use a custom parsing routine to decode the command codes and additional parameters from the C2 before executing them.[263]

S1148 Raccoon Stealer

Raccoon Stealer uses RC4-encrypted, base64-encoded strings to obfuscate functionality and command and control servers.[264][265]

S0565 Raindrop

Raindrop decrypted its Cobalt Strike payload using an AES-256 encryption algorithm in CBC mode with a unique key per sample.[266][267]

S0629 RainyDay

RainyDay can decrypt its payload via a XOR key.[268]

S0458 Ramsay

Ramsay can extract its agent from the body of a malicious document.[269]

S1212 RansomHub

RansomHub can use a provided passphrase to decrypt its configuration file.[270]

S1113 RAPIDPULSE

RAPIDPULSE listens for specific HTTP query parameters in received communications. If specific parameters match, a hard-coded RC4 key is used to decrypt the HTTP query paremter `hmacTime`. This decrypts to a filename that is then open, read, encrypted with the same RC4 key, base64-encoded, written to standard out, then passed as a response to the HTTP request.[271]

S1130 Raspberry Robin

Raspberry Robin contains several layers of obfuscation to hide malicious code from detection and analysis.[272]

S0495 RDAT

RDAT can deobfuscate the base64-encoded and AES-encrypted files downloaded from the C2 server.[273]

S1240 RedLine Stealer

RedLine Stealer has decoded its payload prior to execution.[274]

C0056 RedPenguin

During RedPenguin, UNC3886 used malware implants to deobfuscate incoming C2 messages and encoded archives.[275][276]

S0511 RegDuke

RegDuke can decrypt strings with a key either stored in the Registry or hardcoded in the code.[113]

S0375 Remexi

Remexi decrypts the configuration data using XOR with 25-character keys.[277]

S1219 REPTILE

The REPTILE launcher component can decrypt kernel module code from a file and load it into memory.[212]

S0496 REvil

REvil can decode encrypted strings to enable execution of commands and payloads.[278][279][280][281][282][283]

S0258 RGDoor

RGDoor decodes Base64 strings and decrypts strings using a custom XOR algorithm.[284]

S1222 RIFLESPINE

RIFLESPINE can deobfuscate encrypted files prior to execution on targeted hosts.[212]

S0448 Rising Sun

Rising Sun has decrypted itself using a single-byte XOR scheme. Additionally, Rising Sun can decrypt its configuration data at runtime.[285]

S1150 ROADSWEEP

ROADSWEEP can decrypt embedded scripts prior to execution.[64][286]

G0106 Rocke

Rocke has extracted tar.gz files after downloading them from a C2 server.[287]

S0270 RogueRobin

RogueRobin decodes an embedded executable using base64 and decompresses it.[288]

S0240 ROKRAT

ROKRAT can decrypt strings using the victim's hostname as the key.[289][290]

S1078 RotaJakiro

RotaJakiro uses the AES algorithm, bit shifts in a function called `rotate`, and an XOR cipher to decrypt resources required for persistence, process guarding, and file locking. It also performs this same function on encrypted stack strings and the `head` and `key` sections in the network packet structure used for C2 communications.[291]

S1210 Sagerunex

Sagerunex uses a custom decryption routine to unpack itself during installation.[292]

S1018 Saint Bot

Saint Bot can deobfuscate strings and files for execution.[293]

S1168 SampleCheck5000

SampleCheck5000 can decode and decrypt command line strings and files received through C2.[161][228]

G0034 Sandworm Team

Sandworm Team's VBS backdoor can decode Base64-encoded data and save it to the %TEMP% folder. The group also decrypted received information using the Triple DES algorithm and decompresses it using GZip.[294][295]

S1085 Sardonic

Sardonic can first decrypt with the RC4 algorithm using a hardcoded decryption key before decompressing.[296]

S0461 SDBbot

SDBbot has the ability to decrypt and decompress its payload to enable code execution.[297][298]

S0596 ShadowPad

ShadowPad has decrypted a binary blob to start execution.[299]

S0140 Shamoon

Shamoon decrypts ciphertext using an XOR cipher and a base64-encoded string.[300]

C0058 SharePoint ToolShell Exploitation

During SharePoint ToolShell Exploitation, threat actors decrypted scripts prior to execution.[301]

S1019 Shark

Shark can extract and decrypt downloaded .zip files.[302]

S0546 SharpStage

SharpStage has decompressed data received from the C2 server.[303]

S0444 ShimRat

ShimRat has decompressed its core DLL using shellcode once an impersonated antivirus component was running on a system.[304]

S0589 Sibot

Sibot can decrypt data received from a C2 and save to a file.[130]

S0610 SideTwist

SideTwist can decode and decrypt messages received from C2.[305]

S0623 Siloscape

Siloscape has decrypted the password of the C2 server with a simple byte by byte XOR. Siloscape also writes both an archive of Tor and the `unzip` binary to disk from data embedded within the payload using Visual Studio's Resource Manager.[306]

S0468 Skidmap

Skidmap has the ability to download, unpack, and decrypt tar.gz files .[307]

S1110 SLIGHTPULSE

SLIGHTPULSE can deobfuscate base64 encoded and RC4 encrypted C2 messages.[308]

S0226 Smoke Loader

Smoke Loader deobfuscates its code.[309]

S1086 Snip3

Snip3 can decode its second-stage PowerShell script prior to execution.[310]

C0024 SolarWinds Compromise

During the SolarWinds Compromise, APT29 used 7-Zip to decode their Raindrop malware.[266]

S0615 SombRAT

SombRAT can run `upload` to decrypt and upload files from storage.[83][120]

S0516 SoreFang

SoreFang can decode and decrypt exfiltrated data sent to C2.[311]

S0543 Spark

Spark has used a custom XOR algorithm to decrypt the payload.[312]

S1140 Spica

Upon execution Spica can decode an embedded .pdf and write it to the desktop as a decoy document.[313]

S1232 SplatDropper

SplatDropper has decoded XOR encrypted payload.[82]

S0390 SQLRat

SQLRat has scripts that are responsible for deobfuscating additional scripts.[314]

S1030 Squirrelwaffle

Squirrelwaffle has decrypted files and payloads using a XOR-based algorithm.[315][316]

S0188 Starloader

Starloader decrypts and executes shellcode from a file called Stars.jps.[317]

S1227 StarProxy

StarProxy has decrypted network packets using a custom algorithm.[318]

S1112 STEADYPULSE

STEADYPULSE can URL decode key/value pairs sent over C2.[308]

S1200 StealBit

StealBit can deobfuscate loaded modules prior to execution.[184][319]

G1046 Storm-1811

Storm-1811 has distributed password-protected archives such as ZIP files during intrusions.[320]

S1183 StrelaStealer

StrelaStealer payloads have included strings encrypted via XOR.[321] StrelaStealer JavaScript payloads utilize Base64-encoded payloads that are decoded via certutil to create a malicious DLL file.[322][323]

S0603 Stuxnet

Stuxnet decrypts resources that are loaded into memory and executed.[324]

S0562 SUNSPOT

SUNSPOT decrypts SUNBURST, which was stored in AES128-CBC encrypted blobs.[325]

S0663 SysUpdate

SysUpdate can deobfuscate packed binaries in memory.[151]

G0092 TA505

TA505 has decrypted packed DLLs with an XOR key.[326]

S0011 Taidoor

Taidoor can use a stream cipher to decrypt stings used by the malware.[327]

G0139 TeamTNT

TeamTNT has used a script that decodes a Base64-encoded version of WeaveWorks Scope.[328]

S0560 TEARDROP

TEARDROP was decoded using a custom rolling XOR algorithm to execute a customized Cobalt Strike payload.[329][330][267]

S1223 THINCRUST

THINCRUST can deobfuscate RSA encrypted C2 commands received through the DEVICEID cookie.[61]

G0027 Threat Group-3390

During execution, Threat Group-3390 malware deobfuscates and decompresses code that was encoded with Metasploit's shikata_ga_nai encoder as well as compressed with LZNT1 compression.[331]

S0665 ThreatNeedle

ThreatNeedle can decrypt its payload using RC4, AES, or one-byte XORing.[332]

S1239 TONESHELL

TONESHELL has decoded its payload prior to execution.[69][333][256][318][334]

S0678 Torisma

Torisma has used XOR and Base64 to decode C2 data.[335]

S0266 TrickBot

TrickBot decodes the configuration data and modules.[336][337][338]

G0081 Tropic Trooper

Tropic Trooper used shellcode with an XOR algorithm to decrypt a payload. Tropic Trooper also decrypted image files which contained a payload.[339][340]

S0436 TSCookie

TSCookie has the ability to decrypt, load, and execute a DLL and its resources.[341]

S0647 Turian

Turian has the ability to use a XOR decryption key to extract C2 server domains and IP addresses.[342]

G0010 Turla

Turla has used a custom decryption routine, which pulls key and salt values from other artifacts such as a WMI filter or PowerShell Profile, to decode encrypted PowerShell payloads.[343]

S0263 TYPEFRAME

One TYPEFRAME variant decrypts an archive using an RC4 key, then decompresses and installs the decrypted malicious DLL module. Another variant decodes the embedded file by XORing it with the value "0x35".[344]

S1164 UPSTYLE

UPSTYLE encodes its main content prior to loading via Python as base64-encoded blobs.[345][346]

S0022 Uroburos

Uroburos can decrypt command parameters sent through C2 and use unpacking code to extract its packed executable.[347]

S0386 Ursnif

Ursnif has used crypto key information stored in the Registry to decrypt Tor clients dropped to disk.[348]

S0476 Valak

Valak has the ability to decode and decrypt downloaded files.[349][350]

S0636 VaporRage

VaporRage can deobfuscate XOR-encoded shellcode prior to execution.[46]

S0257 VERMIN

VERMIN decrypts code, strings, and commands to use once it's on the victim's machine.[351]

S0180 Volgmer

Volgmer deobfuscates its strings and APIs once its executed.[352]

G1017 Volt Typhoon

Volt Typhoon has used Base64-encoded data to transfer payloads and commands, including deobfuscation via certutil.[353]

S0670 WarzoneRAT

WarzoneRAT can use XOR 0x45 to decrypt obfuscated code.[354]

S0612 WastedLocker

WastedLocker's custom cryptor, CryptOne, used an XOR based algorithm to decrypt the payload.[355]

C0037 Water Curupira Pikabot Distribution

Water Curupira Pikabot Distribution used highly obfuscated JavaScript files as one initial installer for Pikabot.[356]

S0579 Waterbear

Waterbear has the ability to decrypt its RC4 encrypted payload for execution.[357]

S0515 WellMail

WellMail can decompress scripts received from C2.[358]

S0514 WellMess

WellMess can decode and decrypt data received from C2.[359][360][361]

S0689 WhisperGate

WhisperGate can deobfuscate downloaded files stored in reverse byte order and decrypt embedded resources using multiple XOR operations.[362][363]

S0466 WindTail

WindTail has the ability to decrypt strings using hard-coded AES keys.[364]

S0430 Winnti for Linux

Winnti for Linux has decoded XOR encoded strings holding its configuration upon execution.[365]

S0141 Winnti for Windows

The Winnti for Windows dropper can decrypt and decompresses a data blob.[366]

G1035 Winter Vivern

Winter Vivern delivered exploit payloads via base64-encoded payloads in malicious email messages.[367]

S1115 WIREFIRE

WIREFIRE can decode, decrypt, and decompress data received in C2 HTTP `POST` requests.[368]

G0090 WIRTE

WIRTE has used Base64 to decode malicious VBS script.[369]

S1065 Woody RAT

Woody RAT can deobfuscate Base64-encoded strings and scripts.[370]

S0653 xCaon

xCaon has decoded strings from the C2 server before executing commands.[371]

S1207 XLoader

XLoader uses XOR and RC4 algorithms to decrypt payloads and functions.[372] XLoader can be distributed as a self-extracting RAR archive that launches an AutoIT loader.[373]

S1248 XORIndex Loader

XORIndex Loader can decode its payload prior to execution.[143]

S0388 YAHOYAH

YAHOYAH decrypts downloaded files before execution.[374]

S0251 Zebrocy

Zebrocy decodes its secondary payload and writes it to the victim’s machine. Zebrocy also uses AES and XOR to decrypt strings and payloads.[375][376]

S0230 ZeroT

ZeroT shellcode decrypts and decompresses its RC4-encrypted payload.[377]

S0330 Zeus Panda

Zeus Panda decrypts strings in the code during the execution process.[378]

G0128 ZIRCONIUM

ZIRCONIUM has used the AES256 algorithm with a SHA1 derived key to decrypt exploit code.[379]

S1013 ZxxZ

ZxxZ has used a XOR key to decrypt strings.[380]
