Aug 09 2024

NETMANAGEIT

Intelligence Report

PureHVNC Deployed via Python Multi-stage Loader

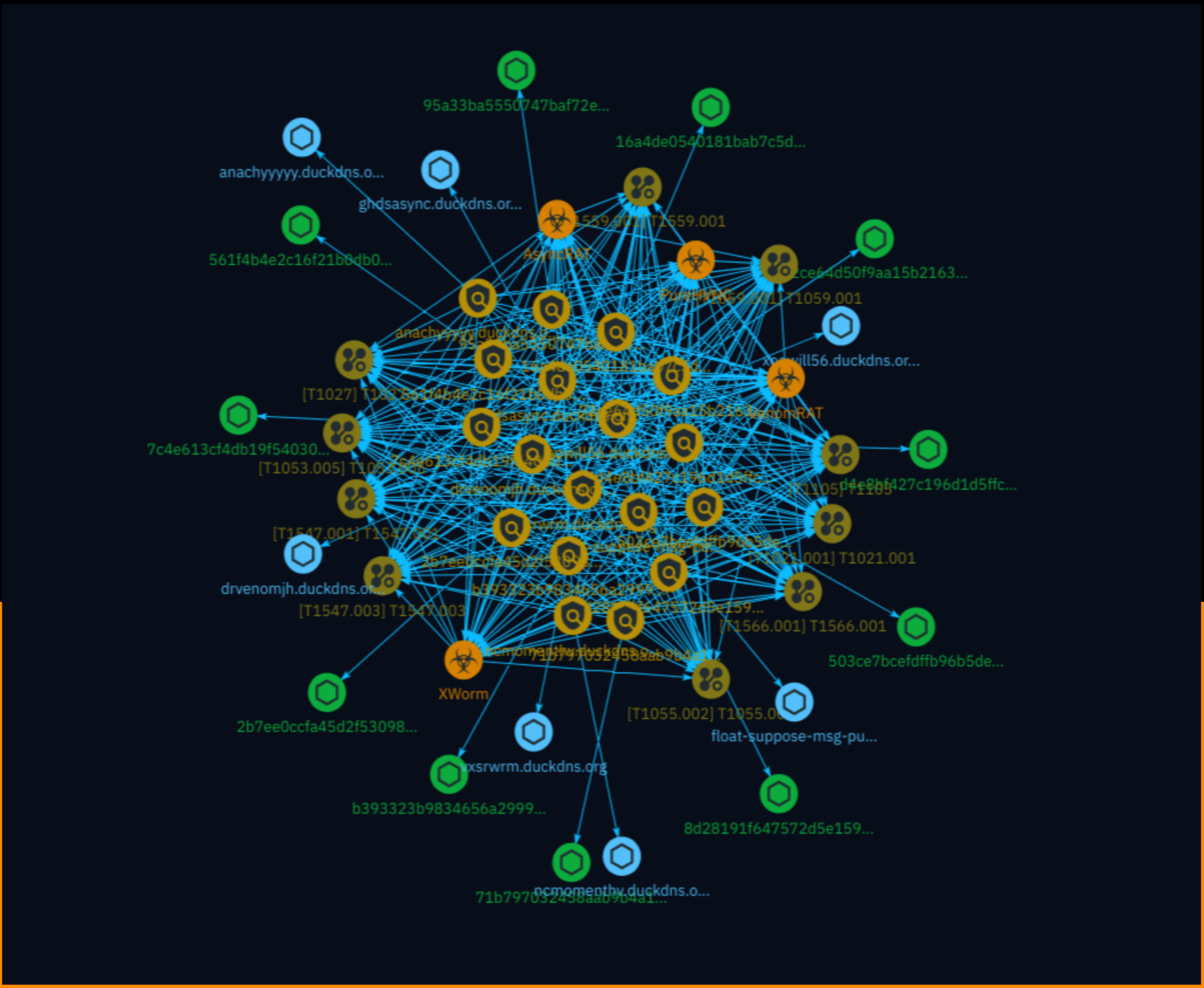

Table of contents

Overview

Entities

Observables

External References

Overview

Description

FortiGuard Labs uncovered a sophisticated attack campaign utilizing multiple obfuscation and evasion techniques to distribute and execute various malware, including VenomRAT, XWorm, AsyncRAT, and PureHVNC. The campaign starts with a phishing email containing a malicious attachment that initiates a series of harmful activities. All the malware employs packing and obfuscation tools like Kramer, donut, and laZzzy to conceal their presence. The analysis focuses on the PureHVNC malware, which collects victim information, targets crypto wallets, password managers, and two-factor authenticators, and can execute additional plugins for remote desktop control and execution.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

100 / 100

Content

N/A

Attack-Pattern

Name

T1059.001

ID

T1059.001

Description

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems). PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk. A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), [PowerSploit](https://attack.mitre.org/software/S0194), [PoshC2](https://attack.mitre.org/software/S0378), and PSAttack.(Citation: Github PSAttack) PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

Name

T1053.005

ID

T1053.005

Description

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The [schtasks](https://attack.mitre.org/software/S0111) utility can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows netapi32 library to create a scheduled task. The deprecated [at](https://attack.mitre.org/software/S0110) utility could also be abused by adversaries (ex: [At](https://attack.mitre.org/techniques/T1053/002)), though `at.exe` can not access tasks created with `schtasks` or the Control Panel. An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and/or to run a process under the context of a specified account (such as SYSTEM). Similar to [System Binary Proxy Execution](https://attack.mitre.org/techniques/T1218), adversaries have also abused the Windows Task Scheduler to potentially mask one-time execution under signed/trusted system processes.(Citation: ProofPoint Serpent) Adversaries may also create "hidden" scheduled tasks (i.e. [Hide Artifacts](https://attack.mitre.org/techniques/T1564)) that may not be visible to defender tools and manual queries used to enumerate tasks. Specifically, an adversary may hide a task from `schtasks /query` and the Task Scheduler by deleting the associated Security Descriptor (SD) registry value (where deletion of this value must be completed using SYSTEM permissions).(Citation: SigmaHQ)(Citation: Tarrask scheduled task) Adversaries may also employ alternate methods to hide tasks, such as altering the metadata (e.g., `Index` value) within associated registry keys.(Citation: Defending Against Scheduled Task Attacks in Windows Environments)

Name

T1566.001

ID

T1566.001

Description

Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon [User Execution](https://attack.mitre.org/techniques/T1204) to gain execution. (Citation: Unit 42 DarkHydrus July 2018) Spearphishing may also involve social engineering techniques, such as posing as a trusted source. There are many options for the attachment such as Microsoft Office documents, executables, PDFs, or archived files. Upon opening the attachment (and potentially clicking past protections), the adversary's payload exploits a vulnerability or directly executes on the user's system. The text of the spearphishing email usually tries to give a plausible reason why the file should be opened, and may explain how to bypass system protections in order to do so. The email may also contain instructions on how to decrypt an attachment, such as a zip file password, in order to evade email boundary defenses. Adversaries frequently manipulate file extensions and icons in order to make attached executables appear to be document files, or files exploiting one application appear to be a file for a different one.

Name

T1547.003

ID

T1547.003

Description

Adversaries may abuse time providers to execute DLLs when the system boots. The Windows Time service (W32Time) enables time synchronization across and within domains. (Citation: Microsoft W32Time Feb 2018) W32Time time providers are responsible for retrieving time stamps from hardware/network resources and outputting these values to other network clients.(Citation: Microsoft TimeProvider) Time providers are implemented as dynamic-link libraries (DLLs) that are registered in the subkeys of

`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\TimeProviders\`. (Citation: Microsoft TimeProvider) The time provider manager, directed by the service control manager, loads and starts time providers listed and enabled under this key at system startup and/or whenever parameters are changed.(Citation: Microsoft TimeProvider) Adversaries may abuse this architecture to establish persistence, specifically by creating a new arbitrarily named subkey pointing to a malicious DLL in the `DllName` value. Administrator privileges are required for time provider registration, though execution will run in context of the Local Service account.(Citation: Github W32Time Oct 2017)

Name

T1027

ID

T1027

Description

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

Name

T1105

ID

T1105

Description

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)). On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil](https://attack.mitre.org/software/S0160), and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as `IEX(New-Object Net.WebClient).downloadString()` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`.(Citation: t1105_lolbas) Adversaries may also abuse installers and package managers, such as `yum` or `winget`, to download tools to victim hosts. Adversaries have also abused file application features, such as the Windows `search-ms` protocol handler, to deliver malicious files to victims through remote file searches invoked by [User Execution](https://attack.mitre.org/techniques/T1204) (typically after interacting with [Phishing](https://attack.mitre.org/techniques/T1566) lures).(Citation: T1105: Trellix_search-ms) Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016) In some cases, adversaries may be able to leverage services that sync between a web-based and an on-premises client, such as Dropbox or OneDrive, to transfer files onto victim systems. For example, by compromising a cloud account and logging into the service's web portal, an adversary may be able to trigger an automatic syncing process that transfers the file onto the victim's machine.(Citation: Dropbox Malware Sync)

Name

T1055.002

ID

T1055.002

Description

Adversaries may inject portable executables (PE) into processes in order to evade process-based defenses as well as possibly elevate privileges. PE injection is a method of executing arbitrary code in the address space of a separate live process. PE injection is commonly performed by copying code (perhaps without a file on disk) into the virtual address space of the target process before invoking it via a new thread. The write can be performed with native Windows API calls such as `VirtualAllocEx` and `WriteProcessMemory`, then invoked with `CreateRemoteThread` or additional code (ex: shellcode). The displacement of the injected code does introduce the additional requirement for functionality to remap memory references. (Citation: Elastic Process Injection July 2017) Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via PE injection may also evade detection from security products since the execution is masked under a legitimate process.

Name

T1547.001

ID

T1547.001

Description

Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. (Citation: Microsoft Run Key) These programs will be executed under the context of the user and will have the account's associated permissions level. The following run keys are created by default on Windows systems: *

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce` Run keys may exist under multiple hives.(Citation: Microsoft Wow6432Node 2018)(Citation: Malwarebytes Wow6432Node 2016) The `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx` is also available but is not created by default on Windows Vista and newer. Registry run key entries can reference programs directly or list them as a dependency.(Citation: Microsoft Run Key) For example, it is possible to load a DLL at logon using a "Depend" key with RunOnceEx: `reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend /v 1 /d "C:\temp\evil[.]dll"` (Citation: Oddvar Moe RunOnceEx Mar 2018) Placing a program within a startup folder will also cause that program to execute when a user logs in. There is a startup folder location for individual user accounts as well as a system-wide startup folder that will be checked regardless of which user account logs in. The startup folder path for the current user is `C:\Users\\[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`. The startup folder path for all users is `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp`. The following Registry keys can be used to set startup folder items for persistence: * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders` * `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders` * `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders` The following Registry keys can control automatic startup of services during boot: * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices` Using policy settings to specify startup programs creates corresponding values in either of two Registry keys: * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run` Programs listed in the load value of the registry key `HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows` run automatically for the currently logged-on user. By default, the multistring `BootExecute` value of the registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager` is set to `autocheck autochk *`. This value causes Windows, at startup, to check the file-system integrity of the hard disks if the system has been shut down abnormally. Adversaries can add other programs or processes to this registry value which will automatically launch at boot. Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use [Masquerading](https://attack.mitre.org/techniques/T1036) to make the Registry entries look as if they are associated with legitimate programs.

Name

T1021.001

ID

T1021.001

Description

Adversaries may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to log into a computer using the Remote Desktop Protocol (RDP). The adversary may then perform actions as the logged-on user. Remote desktop is a common feature in operating systems. It allows a user to log into an interactive session with a system desktop graphical user interface on a remote system. Microsoft refers to its implementation of the Remote Desktop Protocol (RDP) as Remote Desktop Services (RDS).(Citation: TechNet Remote Desktop Services) Adversaries may connect to a remote system over RDP/RDS to expand access if the service is enabled and allows access to accounts with known credentials. Adversaries will likely use Credential Access techniques to acquire credentials to use with RDP. Adversaries may also use RDP in conjunction with the [Accessibility Features](https://attack.mitre.org/techniques/T1546/008) or [Terminal Services DLL](https://attack.mitre.org/techniques/T1505/005) for Persistence.(Citation: Alperovitch Malware)

Name

T1559.001

ID

T1559.001

Description

Adversaries may use the Windows Component Object Model (COM) for local code execution. COM is an inter-process communication (IPC) component of the native Windows application programming interface (API) that enables interaction between software objects, or executable code that implements one or more interfaces.(Citation: Fireeye Hunting COM June 2019) Through COM, a client object can call methods of server objects, which are typically binary Dynamic Link Libraries (DLL) or executables (EXE).(Citation: Microsoft COM) Remote COM execution is facilitated by [Remote Services](https://attack.mitre.org/techniques/T1021) such as [Distributed Component Object Model](https://attack.mitre.org/techniques/T1021/003) (DCOM).(Citation: Fireeye Hunting COM June 2019) Various COM interfaces are exposed that can be abused to invoke arbitrary execution via a variety of programming languages such as C, C++, Java, and [Visual Basic](https://attack.mitre.org/techniques/T1059/005).(Citation: Microsoft COM) Specific COM objects also exist to directly perform functions beyond code execution, such as creating a [Scheduled Task/Job](https://attack.mitre.org/techniques/T1053), fileless download/execution, and other adversary behaviors related to privilege escalation and persistence.(Citation: Fireeye Hunting COM June 2019)(Citation: ProjectZero File Write EoP Apr 2018)

Indicator

Name

2b7ee0ccfa45d2f53098cd8aa4ce73cb00ace462d8490e6843bf05cd07854553

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '2b7ee0ccfa45d2f53098cd8aa4ce73cb00ace462d8490e6843bf05cd07854553']

Name

503ce7bcefdffb96b5de78254f947598a410b86d3aaf597c7334e248c46dae5b

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '503ce7bcefdffb96b5de78254f947598a410b86d3aaf597c7334e248c46dae5b']

Name

16a4de0540181bab7c5d25fcdf90838a28f2dff4ed9e0e37de3f5f1ab20afe0a

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '16a4de0540181bab7c5d25fcdf90838a28f2dff4ed9e0e37de3f5f1ab20afe0a']

Name

drvenomjh.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'drvenomjh.duckdns.org']

Name

float-suppose-msg-pulling.trycloudflare.com

Pattern Type

stix

Pattern

[hostname:value = 'float-suppose-msg-pulling.trycloudflare.com']

Name
ncmomenthv.duckdns.org
Pattern Type
stix
Pattern
[hostname:value = 'ncmomenthv.duckdns.org']
Name
b393323b9834656a2999198d4f02c1a159c6034d3c20c483d22a30aab3810c0c
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'b393323b9834656a2999198d4f02c1a159c6034d3c20c483d22a30aab3810c0c']
Name
d4e8bf427c196d1d5ffca52a5af7162cc5cf4df730ee3fe65b4381ac79662a15
Pattern Type
stix
Pattern

[file:hashes.'SHA-256' = 'd4e8bf427c196d1d5ffca52a5af7162cc5cf4df730ee3fe65b4381ac79662a15']

Name

95a33ba5550747baf72e39b020e6215b6047983eda17250408cd6f4c16a93089

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '95a33ba5550747baf72e39b020e6215b6047983eda17250408cd6f4c16a93089']

Name

vxsrwrm.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'vxsrwrm.duckdns.org']

Name

7c4e613cf4db19f54030097687227809f965a951a26a44a882692ece6e642e3c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7c4e613cf4db19f54030097687227809f965a951a26a44a882692ece6e642e3c']

Name

ghdsasync.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'ghdsasync.duckdns.org']

Name

561f4b4e2c16f21b0db015819340fc59484e4994022c4cca46cf778006d5d441

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '561f4b4e2c16f21b0db015819340fc59484e4994022c4cca46cf778006d5d441']

Name

8d28191f647572d5e159f35ae55120ddf56209a18f2ca95a28d3ca9408b90d68

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'8d28191f647572d5e159f35ae55120ddf56209a18f2ca95a28d3ca9408b90d68']

Name

xoowill56.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'xoowill56.duckdns.org']

Name

anachyyyyy.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'anachyyyyy.duckdns.org']

Name

71b797032458aab9b4a1a203e7ca413f009af1961cffb98590e34f672574599a

Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '71b797032458aab9b4a1a203e7ca413f009af1961cffb98590e34f672574599a']
Name
72ce64d50f9aa15b21631307d2143f426364634a7a2ee4b401ef76bd88c4ff3b
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '72ce64d50f9aa15b21631307d2143f426364634a7a2ee4b401ef76bd88c4ff3b']

Malware

Name
VenomRAT

Name
PureHVNC

Name
XWorm

Name
AsyncRAT

indicates

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name
Name
Name
Name
Name
Name
Name
Name
Name
Name
Name

Name
Name
Name
Name
Name
Name
Name
Name
Name
Name
Name

Name

Name

Name

Name

Name

Name

based-on

Name
Name
Name
Name

uses

Name

Name

Name

Name

Name

Name

Name

Name

Name

StixFile

Value
95a33ba5550747baf72e39b020e6215b6047983eda17250408cd6f4c16a93089
b393323b9834656a2999198d4f02c1a159c6034d3c20c483d22a30aab3810c0c
d4e8bf427c196d1d5ffca52a5af7162cc5cf4df730ee3fe65b4381ac79662a15
16a4de0540181bab7c5d25fcdf90838a28f2dff4ed9e0e37de3f5f1ab20afe0a
71b797032458aab9b4a1a203e7ca413f009af1961cffb98590e34f672574599a
2b7ee0ccfa45d2f53098cd8aa4ce73cb00ace462d8490e6843bf05cd07854553
561f4b4e2c16f21b0db015819340fc59484e4994022c4cca46cf778006d5d441
8d28191f647572d5e159f35ae55120ddf56209a18f2ca95a28d3ca9408b90d68
72ce64d50f9aa15b21631307d2143f426364634a7a2ee4b401ef76bd88c4ff3b
503ce7bcefdffb96b5de78254f947598a410b86d3aaf597c7334e248c46dae5b
7c4e613cf4db19f54030097687227809f965a951a26a44a882692ece6e642e3c

Hostname

Value
xoowill56.duckdns.org
ncmomenthv.duckdns.org
drvenomjh.duckdns.org
vxsrwrm.duckdns.org
anachyyyyy.duckdns.org
float-suppose-msg-pulling.trycloudflare.com
ghdsasync.duckdns.org

External References

- https://www.fortinet.com/blog/threat-research/purehvnc-deployed-via-python-multi-stage-loader
- https://otx.alienvault.com/pulse/66b5fc98a95a09dfa8847b51