Apr 14 2024

NETMANAGEIT

Intelligence Report

VenomRAT Deployed with Arsenal of Plugins

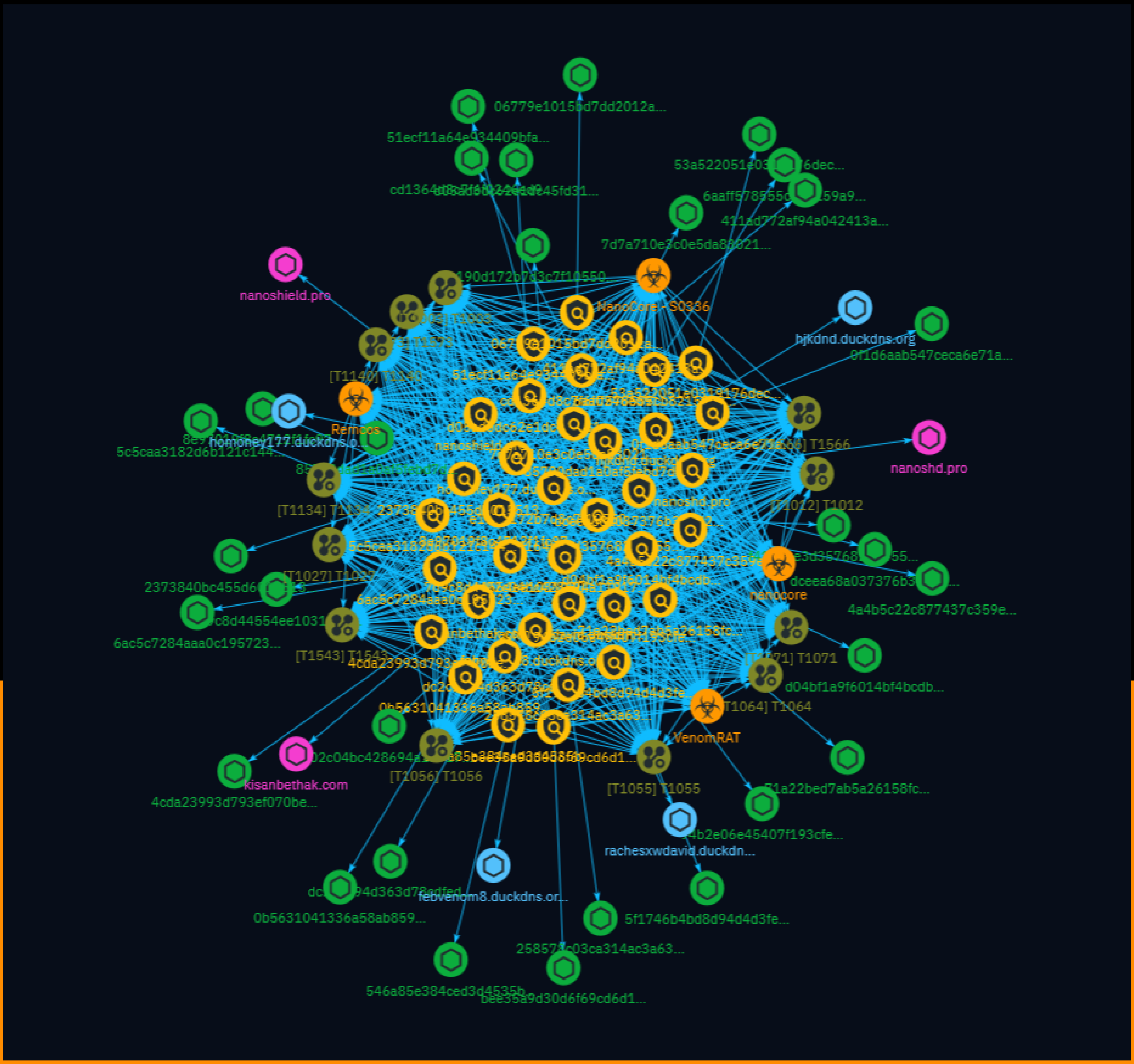

Table of contents

Overview

Entities

Observables

External References

Overview

Description

This report provides an analysis of a recent phishing campaign distributing VenomRAT malware using multiple obfuscation techniques including ScrubCrypt batch files. The attackers send emails with SVG attachments to drop ZIP files containing obfuscated batch scripts. ScrubCrypt is used to decrypt and load VenomRAT, which retrieves additional plugins like Remcos and NanoCore from its C2 server. The campaign shows the threat actors' ability to evade detection and persist in victim systems.

Confidence

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

100 / 100

Content

N/A

Indicator

Name

rachesxwdavid.duckdns.org

Pattern Type

stix

Pattern

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

Name

homoney177.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'homoney177.duckdns.org']

Name

hjkdnd.duckdns.org

Pattern Type

stix

Pattern

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

Name

febvenom8.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'febvenom8.duckdns.org']

Name

nanoshield.pro

Pattern Type

stix

Pattern

[domain-name:value = 'nanoshield.pro']

Name

nanoshd.pro

Pattern Type

stix

Pattern

[domain-name:value = 'nanoshd.pro']

Name

kisanbethak.com

Pattern Type

stix

Pattern

[domain-name:value = 'kisanbethak.com']

Name

f4164be3d357682754559aa32ea74c284eee64140d3f56a63a225d5de10d051c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'f4164be3d357682754559aa32ea74c284eee64140d3f56a63a225d5de10d051c']

Name

f02c04bc428694a11917375f41ecb7c7aa326cf242b4c56ed1e7b3ae14d5dd68

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'f02c04bc428694a11917375f41ecb7c7aa326cf242b4c56ed1e7b3ae14d5dd68']

Name

e190d172b7d3c7f1055052f0ed3da5d5979a8a2b622ca2fbcea90774a5bf6008

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e190d172b7d3c7f1055052f0ed3da5d5979a8a2b622ca2fbcea90774a5bf6008']

Name

dceea68a037376b323d2a934f9fdc59bfbd2c2c0ed66014bdf059f403f4dc6f2

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'dceea68a037376b323d2a934f9fdc59bfbd2c2c0ed66014bdf059f403f4dc6f2']

Name

dc2c1694d363d78cdfed0574cf51413b9b48d932e076033bb76cf69a4470b7e9

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'dc2c1694d363d78cdfed0574cf51413b9b48d932e076033bb76cf69a4470b7e9']

Name

d05ad3dc62e1dc45fd31dc2382c1ea5e5f26f4f7692cb2ef8fd1c6e74b69fa16

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'd05ad3dc62e1dc45fd31dc2382c1ea5e5f26f4f7692cb2ef8fd1c6e74b69fa16']

Name

d04bf1a9f6014bf4bcdb3ac4eb6d85bcc4159ae25a7f00c4493cbcb8e892e159

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'd04bf1a9f6014bf4bcdb3ac4eb6d85bcc4159ae25a7f00c4493cbcb8e892e159']

Name

bee35a9d30d6f69cd6d173c6a6a93110cac59ab3710e32eced6f266581e88b87

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'bee35a9d30d6f69cd6d173c6a6a93110cac59ab3710e32eced6f266581e88b87']

Name

cd1364d8c7f6f0246ed91cd294e2e506e7c94ba2f9a33c373c6fcfe04bbe17e7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'cd1364d8c7f6f0246ed91cd294e2e506e7c94ba2f9a33c373c6fcfe04bbe17e7']

Name

94b2e06e45407f193cfe58e18f5c250bbd1b8e857a754f1c366913129b9dada7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '94b2e06e45407f193cfe58e18f5c250bbd1b8e857a754f1c366913129b9dada7']

Name

8e97019f8c4712f1fc9728c4706112a5ef85a05aa809985709faef951925e094

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '8e97019f8c4712f1fc9728c4706112a5ef85a05aa809985709faef951925e094']

Name

85790dad1a0af5febd7d90e0ec9ce680ec87dcc31a94a25bfb454bb121164bfd

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '85790dad1a0af5febd7d90e0ec9ce680ec87dcc31a94a25bfb454bb121164bfd']

Name

7d9c8d44554ee10310805920afb51249a1e8cd3e32b430e8c9638fec316913d3

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7d9c8d44554ee10310805920afb51249a1e8cd3e32b430e8c9638fec316913d3']

Name

7d7a710e3c0e5da830213f9b72f44a72d721adcf17abc838f28286dde8a1e8d9

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7d7a710e3c0e5da830213f9b72f44a72d721adcf17abc838f28286dde8a1e8d9']

Name

71a22bed7ab5a26158fc1cf1b7bb87146254672483aad72736817ff16e656c7b

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '71a22bed7ab5a26158fc1cf1b7bb87146254672483aad72736817ff16e656c7b']

Name

6ac5c7284aaa0c195723df7a78ae610a7ee096b3b5bc19f6838451acd438116e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6ac5c7284aaa0c195723df7a78ae610a7ee096b3b5bc19f6838451acd438116e']

Name

6aaff578555cb82159a9c16a159f0437c39b673744e0c537c4b7f0f67f56c5d9

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6aaff578555cb82159a9c16a159f0437c39b673744e0c537c4b7f0f67f56c5d9']

Name

5f1746b4bd8d94d4d3feb1e2d4a829b6c3bab9341e272341f4b3a1da01d20745

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'5f1746b4bd8d94d4d3feb1e2d4a829b6c3bab9341e272341f4b3a1da01d20745']

Name

5c5caa3182d6b121c1445d6ca81134ec262cd5ea4f9ef1944f993b63d1987647

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'5c5caa3182d6b121c1445d6ca81134ec262cd5ea4f9ef1944f993b63d1987647']

Name

53a522051e0319176dece493b7e2543135ed41c402adbfeda32a5f6be7d68175

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'53a522051e0319176dece493b7e2543135ed41c402adbfeda32a5f6be7d68175']

Name

546a85e384ced3d4535bad16a877ecd36a79849c379c5daa357689116f042c1b

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'546a85e384ced3d4535bad16a877ecd36a79849c379c5daa357689116f042c1b']

Name

51ecf11a64e934409bfada2b6f0c4d89c3420ca95640bc88f928906e6f0b4832

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'51ecf11a64e934409bfada2b6f0c4d89c3420ca95640bc88f928906e6f0b4832']

Name

4a4b5c22c877437c359ef2acaeeb059881da43b11798581cf2f31c2c83fc3418

Pattern Type

stix
Pattern
[file:hashes.'SHA-256' = '4a4b5c22c877437c359ef2acaeeb059881da43b11798581cf2f31c2c83fc3418']
Name
4cda23993d793ef070be7b9066f31a45b10c1e72d809f4a43726da977a0069d8
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '4cda23993d793ef070be7b9066f31a45b10c1e72d809f4a43726da977a0069d8']
Name
411ad772af94a042413af482a2ef356d3217bcc5123353e3c574347cb93e3d5a
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '411ad772af94a042413af482a2ef356d3217bcc5123353e3c574347cb93e3d5a']
Name

258578c03ca314ac3a636a91e8b3245230eae974cf50799d89b3f931e637014c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'258578c03ca314ac3a636a91e8b3245230eae974cf50799d89b3f931e637014c']

Name

2373840bc455d601551304ec46c281b218e90a91dce3823709c213814636e899

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'2373840bc455d601551304ec46c281b218e90a91dce3823709c213814636e899']

Name

0f1d6aab547ceca6e71ac2e5a54afdaea597318fe7b6ca337f5b92fdff596168

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'0f1d6aab547ceca6e71ac2e5a54afdaea597318fe7b6ca337f5b92fdff596168']

Name

0b5631041336a58ab859d273d76c571dd372220dfa1583b597a2fe5339ad4bf7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'0b5631041336a58ab859d273d76c571dd372220dfa1583b597a2fe5339ad4bf7']

Name

06779e1015bd7dd2012ad03f7bb3f34e8d99d6ca41106f89cb9fb1ec46fe034e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'06779e1015bd7dd2012ad03f7bb3f34e8d99d6ca41106f89cb9fb1ec46fe034e']

Malware

Name
NanoCore - S0336
Name
VenomRAT
Name
Remcos
Name
nanocore
Description
[NanoCore](https://attack.mitre.org/software/S0336) is a modular remote access tool developed in .NET that can be used to spy on victims and steal information. It has been used by threat actors since 2013.(Citation: DigiTrust NanoCore Jan 2017)(Citation: Cofense NanoCore Mar 2018)(Citation: PaloAlto NanoCore Feb 2016)(Citation: Unit 42 Gorgon Group Aug 2018)

Attack-Pattern

Name
T1134

ID
T1134

Description
Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. These token can then be applied to an existing process (i.e. [Token Impersonation/Theft](https://attack.mitre.org/techniques/T1134/001)) or used to spawn a new process (i.e. [Create Process with Token](https://attack.mitre.org/techniques/T1134/002)). An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can then use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.(Citation: Pentestlab Token Manipulation) Any standard user can use the \`runas\` command, and the Windows API functions, to create impersonation tokens; it does not require access to an administrator account. There are also other mechanisms, such as Active Directory fields, that can be used to modify access tokens.

Name

T1012

ID

T1012

Description

Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. The Registry contains a significant amount of information about the operating system, configuration, software, and security.(Citation: Wikipedia Windows Registry) Information can easily be queried using the [Reg](https://attack.mitre.org/software/S0075) utility, though other means to access the Registry exist. Some of the information may help adversaries to further their operation within a network. Adversaries may use the information from [Query Registry](https://attack.mitre.org/techniques/T1012) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Name

T1056

ID

T1056

Description

Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. [Credential API Hooking](https://attack.mitre.org/techniques/T1056/004)) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. [Web Portal Capture](https://attack.mitre.org/techniques/T1056/003)).

Name

T1573

ID

T1573

Description

Adversaries may employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if secret keys are encoded and/or generated within malware samples/configuration files.

Name

T1064

ID

T1064

Description

This technique has been deprecated. Please use [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) where appropriate. Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs. Common scripting languages for Windows include VBScript and [PowerShell](https://attack.mitre.org/techniques/T1086) but could also be in the form of command-line batch scripts. Scripts can be embedded inside Office documents as macros that can be set to execute when files used in [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) and other types of spearphishing are opened. Malicious embedded macros are an alternative means of execution than software exploitation through [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203), where adversaries will rely on macros being allowed or that the user will accept to activate them. Many popular offensive frameworks exist which use forms of scripting for security testers and adversaries alike. Metasploit (Citation: Metasploit_Ref), Veil (Citation: Veil_Ref), and PowerSploit (Citation: Powersploit) are three examples that are popular among penetration testers for exploit and post-compromise operations and include many features for evading defenses. Some adversaries are known to use PowerShell. (Citation: Alperovitch 2014)

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

T1566

ID

T1566

Description

Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a trusted source, as well as evasive techniques such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages (e.g., [Email Hiding Rules](https://attack.mitre.org/techniques/T1564/008)).(Citation: Microsoft OAuth Spam 2022)(Citation: Palo Alto Unit 42 VBA Infostealer 2014) Another way to accomplish this is by forging or spoofing(Citation: Proofpoint-spoof) the identity of the sender which can be used to fool both the human recipient as well as automated security tools.(Citation: cyberproof-double-bounce) Victims may also receive phishing messages that instruct them to call a phone number where they are directed to visit a malicious URL, download malware,(Citation: sygnia Luna Month)(Citation: CISA Remote Monitoring and Management Software) or install adversary-accessible remote management tools onto their computer (i.e., [User Execution](https://attack.mitre.org/techniques/T1204)).(Citation: Unit42 Luna Moth)

Name

T1055

ID

T1055

Description

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. 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 process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

Name

T1140

ID

T1140

Description

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [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)

Name

T1071

ID

T1071

Description

Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

Name

T1543

ID

T1543

Description

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](https://attack.mitre.org/techniques/T1543/004) and [Launch Agent](https://attack.mitre.org/techniques/T1543/001) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

Name

T1003

ID

T1003

Description

Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform [Lateral Movement](https://attack.mitre.org/tactics/TA0008) and access restricted information. Several of the tools mentioned in associated sub-techniques may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.

Hostname

Value
rachesxwdavid.duckdns.org
homoney177.duckdns.org
hjkdnd.duckdns.org
febvenom8.duckdns.org

Domain-Name

Value
nanoshield.pro
nanoshd.pro
kisanbethak.com

StixFile

Value
f4164be3d357682754559aa32ea74c284eee64140d3f56a63a225d5de10d051c
f02c04bc428694a11917375f41ecb7c7aa326cf242b4c56ed1e7b3ae14d5dd68
e190d172b7d3c7f1055052f0ed3da5d5979a8a2b622ca2fbcea90774a5bf6008
dceea68a037376b323d2a934f9fdc59bfbd2c2c0ed66014bdf059f403f4dc6f2
dc2c1694d363d78cdfed0574cf51413b9b48d932e076033bb76cf69a4470b7e9
d05ad3dc62e1dc45fd31dc2382c1ea5e5f26f4f7692cb2ef8fd1c6e74b69fa16
d04bf1a9f6014bf4bcdb3ac4eb6d85bcc4159ae25a7f00c4493cbcb8e892e159
cd1364d8c7f6f0246ed91cd294e2e506e7c94ba2f9a33c373c6fcfe04bbe17e7
bee35a9d30d6f69cd6d173c6a6a93110cac59ab3710e32eced6f266581e88b87
94b2e06e45407f193cfe58e18f5c250bbd1b8e857a754f1c366913129b9dada7
8e97019f8c4712f1fc9728c4706112a5ef85a05aa809985709faef951925e094
85790dad1a0af5febd7d90e0ec9ce680ec87dcc31a94a25bfb454bb121164bfd
7d9c8d44554ee10310805920afb51249a1e8cd3e32b430e8c9638fec316913d3

7d7a710e3c0e5da830213f9b72f44a72d721adcf17abc838f28286dde8a1e8d9
71a22bed7ab5a26158fc1cf1b7bb87146254672483aad72736817ff16e656c7b
6ac5c7284aaa0c195723df7a78ae610a7ee096b3b5bc19f6838451acd438116e
6aaff578555cb82159a9c16a159f0437c39b673744e0c537c4b7f0f67f56c5d9
5f1746b4bd8d94d4d3feb1e2d4a829b6c3bab9341e272341f4b3a1da01d20745
5c5caa3182d6b121c1445d6ca81134ec262cd5ea4f9ef1944f993b63d1987647
546a85e384ced3d4535bad16a877ecd36a79849c379c5daa357689116f042c1b
53a522051e0319176dece493b7e2543135ed41c402adbfeda32a5f6be7d68175
51ecf11a64e934409bfada2b6f0c4d89c3420ca95640bc88f928906e6f0b4832
4cda23993d793ef070be7b9066f31a45b10c1e72d809f4a43726da977a0069d8
4a4b5c22c877437c359ef2acaeeb059881da43b11798581cf2f31c2c83fc3418
411ad772af94a042413af482a2ef356d3217bcc5123353e3c574347cb93e3d5a
258578c03ca314ac3a636a91e8b3245230eae974cf50799d89b3f931e637014c
2373840bc455d601551304ec46c281b218e90a91dce3823709c213814636e899
06779e1015bd7dd2012ad03f7bb3f34e8d99d6ca41106f89cb9fb1ec46fe034e
0f1d6aab547ceca6e71ac2e5a54afdaea597318fe7b6ca337f5b92fdff596168
0b5631041336a58ab859d273d76c571dd372220dfa1583b597a2fe5339ad4bf7

External References

- https://www.fortinet.com/blog/threat-research/scrubcrypt-deploys-venomrat-with-arsenal-of-plugins
- https://otx.alienvault.com/pulse/6616d1d234937278cf7362d0