Nov 30 2023

NETMANAGEIT

Intelligence Report

New SugarGh0st RAT targets Uzbekistan government and South Korea

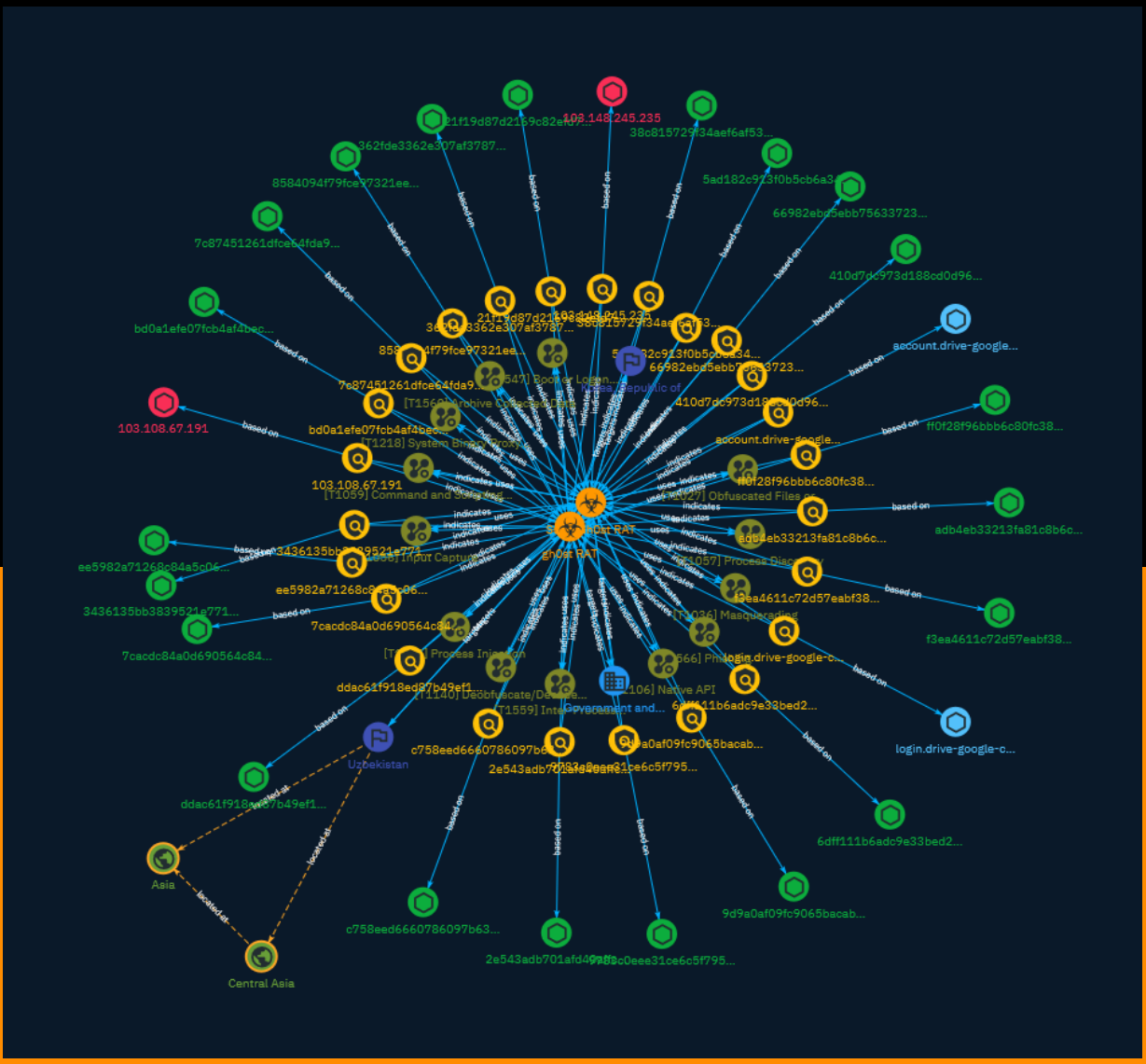

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A suspected Chinese-speaking threat actor is targeting users in Uzbekistan and South Korea, Cisco Talos has found in its analysis of new remote access trojan (RAT) samples.

Confidence

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

15 / 100

Content

N/A

Attack-Pattern

Name

Process Discovery

ID

T1057

Description

Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software/applications running on systems within the network. Adversaries may use the information from [Process Discovery](https://attack.mitre.org/techniques/T1057) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. In Windows environments, adversaries could obtain details on running processes using the [Tasklist](https://attack.mitre.org/software/S0057) utility via [cmd](https://attack.mitre.org/software/S0106) or `Get-Process` via [PowerShell](https://attack.mitre.org/techniques/T1059/001). Information about processes can also be extracted from the output of [Native API](https://attack.mitre.org/techniques/T1106) calls such as `CreateToolhelp32Snapshot`. In Mac and Linux, this is accomplished with the `ps` command. Adversaries may also opt to enumerate processes via /proc. On network devices, [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `show processes` can be used to display current running processes.(Citation: US-CERT-TA18-106A)(Citation: show_processes_cisco_cmd)

Name

Boot or Logon Autostart Execution

ID

T1547

Description

Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon.(Citation: Microsoft Run Key)(Citation: MSDN Authentication Packages)(Citation: Microsoft TimeProvider)(Citation: Cylance Reg Persistence Sept 2013)(Citation: Linux Kernel Programming) These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel. Since some boot or logon autostart programs run with higher privileges, an adversary may leverage these to elevate privileges.

Name

Input Capture

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

Masquerading

ID

T1036

Description

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusable system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](https://attack.mitre.org/techniques/T1090) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.

Name

Process Injection

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

Phishing

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

Native API

ID

T1106

Description

Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. (Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. Native API functions (such as `NtCreateProcess`) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations.(Citation: Microsoft Win32)(Citation: LIBC)(Citation: GLIBC) Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS Cocoa)(Citation: macOS Foundation) Adversaries may use assembly to directly or in-directly invoke syscalls in an attempt to subvert defensive sensors and detection signatures such as user mode API-hooks.(Citation: Redops Syscalls) Adversaries may also attempt to tamper with sensors and defensive tools associated with API monitoring, such as unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001).

Name

Archive Collected Data

ID

T1560

Description

An adversary may compress and/or encrypt data that is collected prior to exfiltration. Compressing the data can help to obfuscate the collected data and minimize the amount of data sent over the network. Encryption can be used to hide information that is being exfiltrated from detection or make exfiltration less conspicuous upon inspection by a defender. Both compression and encryption are done prior to exfiltration, and can be performed using a utility, 3rd party library, or custom method.

Name

Obfuscated Files or Information

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

Inter-Process Communication

ID

T1559

Description

Adversaries may abuse inter-process communication (IPC) mechanisms for local code or command execution. IPC is typically used by processes to share data, communicate with each other, or synchronize execution. IPC is also commonly used to avoid situations such as deadlocks, which occurs when processes are stuck in a cyclic waiting pattern. Adversaries may abuse IPC to execute arbitrary code or commands. IPC mechanisms may differ depending on OS, but typically exists in a form accessible through programming languages/libraries or native interfaces such as Windows [Dynamic Data Exchange](https://attack.mitre.org/techniques/T1559/002) or [Component Object Model](https://attack.mitre.org/techniques/T1559/001). Linux environments support several different IPC mechanisms, two of which being sockets and pipes.(Citation: Linux IPC) Higher level execution mediums, such as those of [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059)s, may also leverage underlying IPC mechanisms. Adversaries may also use [Remote Services](https://attack.mitre.org/techniques/T1021) such as [Distributed Component Object Model](https://attack.mitre.org/techniques/T1021/003) to facilitate remote IPC execution.(Citation: Fireeye Hunting COM June 2019)

Name

Command and Scripting Interpreter

ID

T1059

Description

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python](https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

Name

Deobfuscate/Decode Files or Information

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

System Binary Proxy Execution

ID

T1218

Description

Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system.(Citation: LOLBAS Project) Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands. Similarly, on Linux systems adversaries may abuse trusted binaries such as `split` to proxy execution of malicious commands.(Citation: split man page)(Citation: GTFO split)

Sector

Name
Government and administrations
Description
Civilian government institutions and administrations of the executive and legislative branches. The diplomatic and judicial branches are not included.

Indicator

Name
account.drive-google-com.tk
Pattern Type
stix
Pattern
[hostname:value = 'account.drive-google-com.tk']
Name
7cacdc84a0d690564c8471a4f58ab192ef7d9091ab0809933f616010bbf6846a
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '7cacdc84a0d690564c8471a4f58ab192ef7d9091ab0809933f616010bbf6846a']
Name

5ad182c913f0b5cb6a34126137c335110d4c9472f5c745cb7a438d108b03b27c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '5ad182c913f0b5cb6a34126137c335110d4c9472f5c745cb7a438d108b03b27c']

Name

ff0f28f96bbb6c80fc3823fe71d5e07e1a05b06986e82a2fbe324d68ba5ab2ea

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'ff0f28f96bbb6c80fc3823fe71d5e07e1a05b06986e82a2fbe324d68ba5ab2ea']

Name

3436135bb3839521e7712882f0f6548aff78db66a1064408c49f820a0b85d980

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '3436135bb3839521e7712882f0f6548aff78db66a1064408c49f820a0b85d980']

Name

410d7dc973d188cd0d962a59f48deb1cfc73adf37857765e90194f6e878d4488

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '410d7dc973d188cd0d962a59f48deb1cfc73adf37857765e90194f6e878d4488']

Name

2e543adb701afd40affcb4c51bd8246398b0210bee641ca9aeffcca893c9e4a5

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '2e543adb701afd40affcb4c51bd8246398b0210bee641ca9aeffcca893c9e4a5']

Name

bd0a1efe07fcb4af4bec1b2881a0711f0be34044680ad8cff958a68a70d4a914

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'bd0a1efe07fcb4af4bec1b2881a0711f0be34044680ad8cff958a68a70d4a914']

Name

9783c0eee31ce6c5f795ecf387025af5d55208ff2713c470af2042721ab38606

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '9783c0eee31ce6c5f795ecf387025af5d55208ff2713c470af2042721ab38606']

Name

8584094f79fce97321ee82ca5da41b6830ecc6a0921bcaddb8dd337827cd7d1a

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '8584094f79fce97321ee82ca5da41b6830ecc6a0921bcaddb8dd337827cd7d1a']

Name

103.148.245.235

Description

CC=HK ASN=AS142032 High Family Technology Co., Limited

Pattern Type

stix

Pattern

[ipv4-addr:value = '103.148.245.235']

Name

c758eed6660786097b63ac6748236b5b6084783703ea7ee2111e8f0bcaa3652e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'c758eed6660786097b63ac6748236b5b6084783703ea7ee2111e8f0bcaa3652e']

Name

ee5982a71268c84a5c062095ce135780b8c2ffb1f266c2799173fb0f7bfdd33e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'ee5982a71268c84a5c062095ce135780b8c2ffb1f266c2799173fb0f7bfdd33e']

Name

38c815729f34aef6af531edf3f0c3f09635686dbe7e5db5cb97eca5b2b5b7712

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '38c815729f34aef6af531edf3f0c3f09635686dbe7e5db5cb97eca5b2b5b7712']

Name

login.drive-google-com.tk

Pattern Type

stix

Pattern

[hostname:value = 'login.drive-google-com.tk']

Name

7c87451261dfce64fda987eb395694b5330fd958466c46c931440cd9dc227505

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7c87451261dfce64fda987eb395694b5330fd958466c46c931440cd9dc227505']

Name

f3ea4611c72d57eabf381d5639c3c8d1840cb005ed811f3038410fb2e04978c1

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'f3ea4611c72d57eabf381d5639c3c8d1840cb005ed811f3038410fb2e04978c1']

Name

103.108.67.191

Description

CC=HK ASN=AS142032 High Family Technology Co., Limited

Pattern Type

stix

Pattern

[ipv4-addr:value = '103.108.67.191']

Name

66982ebd5ebb75633723c7057a1e948ac3aafe3ff808397eb0c55c853c82f9e6

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '66982ebd5ebb75633723c7057a1e948ac3aafe3ff808397eb0c55c853c82f9e6']

Name

9d9a0af09fc9065bacabf1a193cad4386b5e8e5101639e07efa82992b723f3b0

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '9d9a0af09fc9065bacabf1a193cad4386b5e8e5101639e07efa82992b723f3b0']

Name

21f19d87d2169c82efd76ddb1baa024a1e59b93f82d28f276de853fc3ef8b20e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '21f19d87d2169c82efd76ddb1baa024a1e59b93f82d28f276de853fc3ef8b20e']

Name

6dff111b6adc9e33bed20eae99bec779f1c29dd55895a71125cfbe3c90950eb2

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6dff111b6adc9e33bed20eae99bec779f1c29dd55895a71125cfbe3c90950eb2']

Name

ddac61f918ed87b49ef15d05873e7f52b919758aef713145f6a7d538c714fa2e

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'ddac61f918ed87b49ef15d05873e7f52b919758aef713145f6a7d538c714fa2e']

Name

362fde3362e307af3787b9bf0b5c71f87b659a3217e054c4d0acea8b9e6d74b0

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '362fde3362e307af3787b9bf0b5c71f87b659a3217e054c4d0acea8b9e6d74b0']

Name

adb4eb33213fa81c8b6cc013a6f4a43fa8b70eb8027433cf4339b532cb6e84cf

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'adb4eb33213fa81c8b6cc013a6f4a43fa8b70eb8027433cf4339b532cb6e84cf']

Country

Name
Uzbekistan

Name
Korea, Republic of

Region

Name
Central Asia

Name
Asia

Malware

Name
SugarGh0st RAT
Name
gh0st RAT
Description
[gh0st RAT](https://attack.mitre.org/software/S0032) is a remote access tool (RAT). The source code is public and it has been used by multiple groups.(Citation: FireEye Hacking Team)(Citation: Arbor Musical Chairs Feb 2018)(Citation: Nccgroup Gh0st April 2018)

StixFile

Value
f3ea4611c72d57eabf381d5639c3c8d1840cb005ed811f3038410fb2e04978c1
9783c0eee31ce6c5f795ecf387025af5d55208ff2713c470af2042721ab38606
adb4eb33213fa81c8b6cc013a6f4a43fa8b70eb8027433cf4339b532cb6e84cf
21f19d87d2169c82efd76ddb1baa024a1e59b93f82d28f276de853fc3ef8b20e
bd0a1efe07fcb4af4bec1b2881a0711f0be34044680ad8cff958a68a70d4a914
8584094f79fce97321ee82ca5da41b6830ecc6a0921bcaddb8dd337827cd7d1a
3436135bb3839521e7712882f0f6548aff78db66a1064408c49f820a0b85d980
362fde3362e307af3787b9bf0b5c71f87b659a3217e054c4d0acea8b9e6d74b0
410d7dc973d188cd0d962a59f48deb1cfc73adf37857765e90194f6e878d4488
7c87451261dfce64fda987eb395694b5330fd958466c46c931440cd9dc227505
66982ebd5ebb75633723c7057a1e948ac3aafe3ff808397eb0c55c853c82f9e6
2e543adb701afd40affcb4c51bd8246398b0210bee641ca9aeffcca893c9e4a5
c758eed6660786097b63ac6748236b5b6084783703ea7ee2111e8f0bcaa3652e

9d9a0af09fc9065bacabf1a193cad4386b5e8e5101639e07efa82992b723f3b0

ee5982a71268c84a5c062095ce135780b8c2ffb1f266c2799173fb0f7bfdd33e

38c815729f34aef6af531edf3f0c3f09635686dbe7e5db5cb97eca5b2b5b7712

5ad182c913f0b5cb6a34126137c335110d4c9472f5c745cb7a438d108b03b27c

ddac61f918ed87b49ef15d05873e7f52b919758aef713145f6a7d538c714fa2e

ff0f28f96bbb6c80fc3823fe71d5e07e1a05b06986e82a2fbe324d68ba5ab2ea

6dff111b6adc9e33bed20eae99bec779f1c29dd55895a71125cfbe3c90950eb2

7cacdc84a0d690564c8471a4f58ab192ef7d9091ab0809933f616010bbf6846a

Hostname

Value
account.drive-google-com.tk
login.drive-google-com.tk

IPv4-Addr

Value
103.148.245.235
103.108.67.191

External References

- https://otx.alienvault.com/pulse/6568b12aaabf4058f1f19eb5
- https://blog.talosintelligence.com/new-sugargh0st-rat/