

NETMANAGEIT

Intelligence Report

RAT Distributed as

UUEncoding (UUE) File

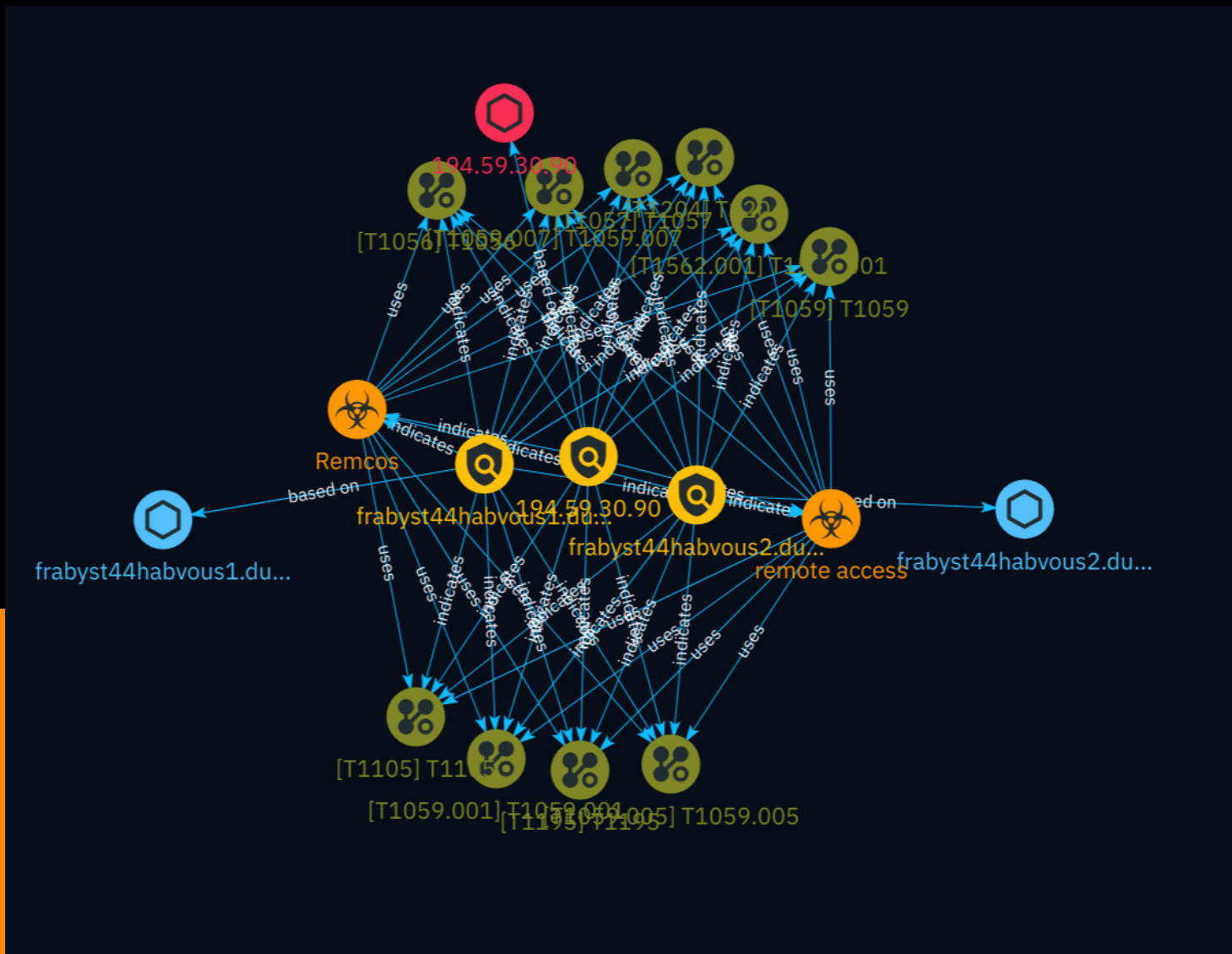


Table of contents

Overview

● Description	4
● Confidence	4
● Content	5

Entities

● Attack-Pattern	6
● Indicator	14
● Malware	16
● based-on	17
● indicates	18
● uses	22

Observables

● Hostname	25
------------	----

● IPv4-Addr	26
-------------	----

External References

● External References	27
-----------------------	----

Overview

Description

This intelligence report describes a malicious operation where the Remcos Remote Access Trojan (RAT) is being disseminated through phishing emails containing an attachment exploiting the Unix-to-Unix Encoding (UUE) technique. The encoded file loads an obfuscated VBScript that fetches additional malicious components, leading to the deployment of the Remcos RAT on compromised systems. The report outlines the multi-stage infection process, providing technical details and indicators of compromise (IOCs) related to this campaign.

Confidence

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

100 / 100

Content

N/A

Attack-Pattern

Name

T1057

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

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

T1059.007

ID

T1059.007

Description

Adversaries may abuse various implementations of JavaScript for execution. JavaScript (JS) is a platform-independent scripting language (compiled just-in-time at runtime) commonly associated with scripts in webpages, though JS can be executed in runtime environments outside the browser.(Citation: NodeJS) JScript is the Microsoft implementation of the same scripting standard. JScript is interpreted via the Windows Script engine and thus integrated with many components of Windows such as the [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and Internet Explorer HTML Application (HTA) pages.(Citation: JScript May 2018)(Citation: Microsoft JScript 2007)(Citation: Microsoft Windows Scripts) JavaScript for Automation (JXA) is a macOS scripting language based on JavaScript, included as part of Apple's Open Scripting Architecture (OSA), that was introduced in OSX 10.10. Apple's OSA provides scripting capabilities to control applications, interface with the operating system, and bridge access into the rest of Apple's internal APIs. As of OSX 10.10, OSA only supports two languages, JXA and [AppleScript](https://attack.mitre.org/techniques/T1059/002). Scripts can be executed via the command line utility `osascript`, they can be compiled into applications or script

files via ``osacompile``, and they can be compiled and executed in memory of other programs by leveraging the OSAKit Framework.(Citation: Apple About Mac Scripting 2016)(Citation: SpecterOps JXA 2020)(Citation: SentinelOne macOS Red Team)(Citation: Red Canary Silver Sparrow Feb2021)(Citation: MDSec macOS JXA and VSCode) Adversaries may abuse various implementations of JavaScript to execute various behaviors. Common uses include hosting malicious scripts on websites as part of a [Drive-by Compromise](https://attack.mitre.org/techniques/T1189) or downloading and executing these script files as secondary payloads. Since these payloads are text-based, it is also very common for adversaries to obfuscate their content as part of [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027).

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

T1204

ID

T1204

Description

An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of [Phishing](https://attack.mitre.org/techniques/T1566). While [User Execution](https://attack.mitre.org/techniques/T1204) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](https://attack.mitre.org/techniques/T1534). Adversaries may also deceive users into performing actions such as enabling [Remote Access Software](https://attack.mitre.org/techniques/T1219), allowing direct control of the system to the adversary, or downloading and executing malware for [User Execution](https://attack.mitre.org/techniques/T1204). For example, tech support scams can be facilitated through [Phishing](https://attack.mitre.org/techniques/T1566), vishing, or various forms of user interaction. Adversaries can use a combination of these methods, such as spoofing and promoting toll-free numbers or call centers that are used to direct victims to malicious websites, to deliver and execute payloads containing malware or [Remote Access Software](https://attack.mitre.org/techniques/T1219).(Citation: Telephone Attack Delivery)

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 `EX(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. 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

T1059

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

T1059.005

ID

T1059.005

Description

Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and the [Native API](https://attack.mitre.org/techniques/T1106) through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.(Citation: VB .NET Mar 2020)(Citation: VB Microsoft) Derivative languages based on VB have also been created, such as Visual Basic for Applications (VBA) and VBScript. VBA is an event-driven programming language built into Microsoft Office, as well as several third-party applications.(Citation: Microsoft VBA) (Citation: Wikipedia VBA) VBA enables documents to contain macros used to automate the execution of tasks and other functionality on the host. VBScript is a default scripting language on Windows hosts and can also be used in place of [JavaScript](https://attack.mitre.org/techniques/T1059/007) on HTML Application (HTA) webpages served to Internet Explorer (though most modern browsers do not come with VBScript support). (Citation: Microsoft VBScript) Adversaries may use VB payloads to execute malicious commands. Common malicious usage includes automating execution of behaviors with VBScript or embedding VBA content into [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001) payloads (which may also involve [Mark-of-the-

Web Bypass](<https://attack.mitre.org/techniques/T1553/005>) to enable execution).(Citation: Default VBS macros Blocking)

Name

T1195

ID

T1195

Description

Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise can take place at any stage of the supply chain including: * Manipulation of development tools * Manipulation of a development environment * Manipulation of source code repositories (public or private) * Manipulation of source code in open-source dependencies * Manipulation of software update/distribution mechanisms * Compromised/infected system images (multiple cases of removable media infected at the factory)(Citation: IBM Storwize)(Citation: Schneider Electric USB Malware) * Replacement of legitimate software with modified versions * Sales of modified/counterfeit products to legitimate distributors * Shipment interdiction While supply chain compromise can impact any component of hardware or software, adversaries looking to gain execution have often focused on malicious additions to legitimate software in software distribution or update channels.(Citation: Avast CCleaner3 2018)(Citation: Microsoft Dofail 2018)(Citation: Command Five SK 2011) Targeting may be specific to a desired victim set or malicious software may be distributed to a broad set of consumers but only move on to additional tactics on specific victims.(Citation: Symantec Elderwood Sept 2012)(Citation: Avast CCleaner3 2018)(Citation: Command Five SK 2011) Popular open source projects that are used as dependencies in many applications may also be targeted as a means to add malicious code to users of the dependency.(Citation: Trendmicro NPM Compromise)

Name

T1562.001

ID

T1562.001

Description

Adversaries may modify and/or disable security tools to avoid possible detection of their malware/tools and activities. This may take many forms, such as killing security software processes or services, modifying / deleting Registry keys or configuration files so that tools do not operate properly, or other methods to interfere with security tools scanning or reporting information. Adversaries may also disable updates to prevent the latest security patches from reaching tools on victim systems.(Citation: SCADafence_ransomware) Adversaries may also tamper with artifacts deployed and utilized by security tools. Security tools may make dynamic changes to system components in order to maintain visibility into specific events. For example, security products may load their own modules and/or modify those loaded by processes to facilitate data collection. Similar to [Indicator Blocking](<https://attack.mitre.org/techniques/T1562/006>), adversaries may unhook or otherwise modify these features added by tools (especially those that exist in userland or are otherwise potentially accessible to adversaries) to avoid detection.(Citation: OutFlank System Calls)(Citation: MDSec System Calls) Adversaries may also focus on specific applications such as Sysmon. For example, the “Start” and “Enable” values in ``HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Microsoft-Windows-Sysmon-Operational`` may be modified to tamper with and potentially disable Sysmon logging.(Citation: disable_win_evt_logging) On network devices, adversaries may attempt to skip digital signature verification checks by altering startup configuration files and effectively disabling firmware verification that typically occurs at boot.(Citation: Fortinet Zero-Day and Custom Malware Used by Suspected Chinese Actor in Espionage Operation)(Citation: Analysis of FG-IR-22-369) In cloud environments, tools disabled by adversaries may include cloud monitoring agents that report back to services such as AWS CloudWatch or Google Cloud Monitor. Furthermore, although defensive tools may have anti-tampering mechanisms, adversaries may abuse tools such as legitimate rootkit removal kits to impair and/or disable these tools.(Citation: chasing_avaddon_ransomware)(Citation: dharma_ransomware)(Citation: demystifying_ryuk)(Citation: doppelpaymer_crowdstrike) For example, adversaries have used tools such as GMER to find and shut down hidden processes and antivirus software on infected systems.(Citation: demystifying_ryuk) Additionally, adversaries may exploit legitimate drivers from anti-virus software to gain access to kernel space (i.e. [Exploitation for Privilege Escalation](<https://attack.mitre.org/techniques/T1068>)), which may lead to bypassing anti-tampering features.(Citation: avoslocker_ransomware)

Indicator

Name

194.59.30.90

Description

```
**ISP:** 12651980 CANADA INC. **OS:** - ----- Services: **22:** ~~~
SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.11 Key type: ssh-rsa Key:
AAAAB3NzaC1yc2EAAAADAQABAAQGCqCqQe+4Hd+KFcEk+MjtG9zdZAJAK+Uvvved8eqJwS3zVD
d
eqOu1aDk277bSYt1CDujTQQxgUOK8pNR10JV8AuPpBgBw4Z17+dCoOLrVcURxDLqwb7YHrrUffwa
8andNi6v20tiazAYtBRlrlPEk2K5q1Zu588Ey/8Fj5vFPiYL5U6KOZh83ynZqObcfvluKZhSeseX
mUfZT2dN54HvCsDg7EFyW0a2svRlyfgWLGmBVg4/69XUMHlkmgcvSC9Vv64vFI+s/3o/1lVhgN
60oXz87KgGlcljcFHwYldriHr+YsSREY1JEBog4WRWxYJ+ZLuW1E9P3dBMD/nA9iV93+HjV6mwl4
Zgv7DpGfEoy5zfAh+KE24/FimGtFyT0fFCfj2ovX1HWooDsuUtu9d8LwoovSvYw70d25tDLAJ9Ws
nW4okNl5Eb03Aji8yUyuaEJqyLuEutZ5C1hOhT+lsGCz+FjbyBxu2IUEDgfAotHXk3DJEEBya8+0
n8gmGaqAYjs= Fingerprint: d7:4f:91:37:94:13:4d:ce:37:29:74:c7:0e:82:df:46 Kex Algorithms:
curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384
ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512
diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 kex-strict-s-
v00@openssh.com Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-
sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com
aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com
MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-
sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-
etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256
hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ~~~
----- **80:** ~~~ HTTP/1.1 200 OK Date: Sat, 08 Jun 2024 21:30:45 GMT Server:
Apache/2.4.41 (Ubuntu) Last-Modified: Thu, 02 May 2024 18:31:10 GMT ETag:
"2aa6-6177ccea24b9" Accept-Ranges: bytes Content-Length: 10918 Vary: Accept-Encoding
Content-Type: text/html ~~~ -----
```

Pattern Type

stix

Pattern

[ipv4-addr:value = '194.59.30.90']

Name

frabyst44habvous1.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'frabyst44habvous1.duckdns.org']

Name

frabyst44habvous2.duckdns.org

Pattern Type

stix

Pattern

[hostname:value = 'frabyst44habvous2.duckdns.org']

Malware

Name

remote access

Name

Remcos

based-on

Name
Name
Name

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

uses

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

TLP:CLEAR



Hostname

Value

frabyst44habvous1.duckdns.org

frabyst44habvous2.duckdns.org

IPv4-Addr

Value

194.59.30.90

External References

-
- <https://asec.ahnlab.com/en/66463/>
-
- <https://otx.alienvault.com/pulse/666822c92c8ceaf905ebcc09>