MATA: Multi-platform targeted malware framework

securelist.com/mata-multi-platform-targeted-malware-framework/97746

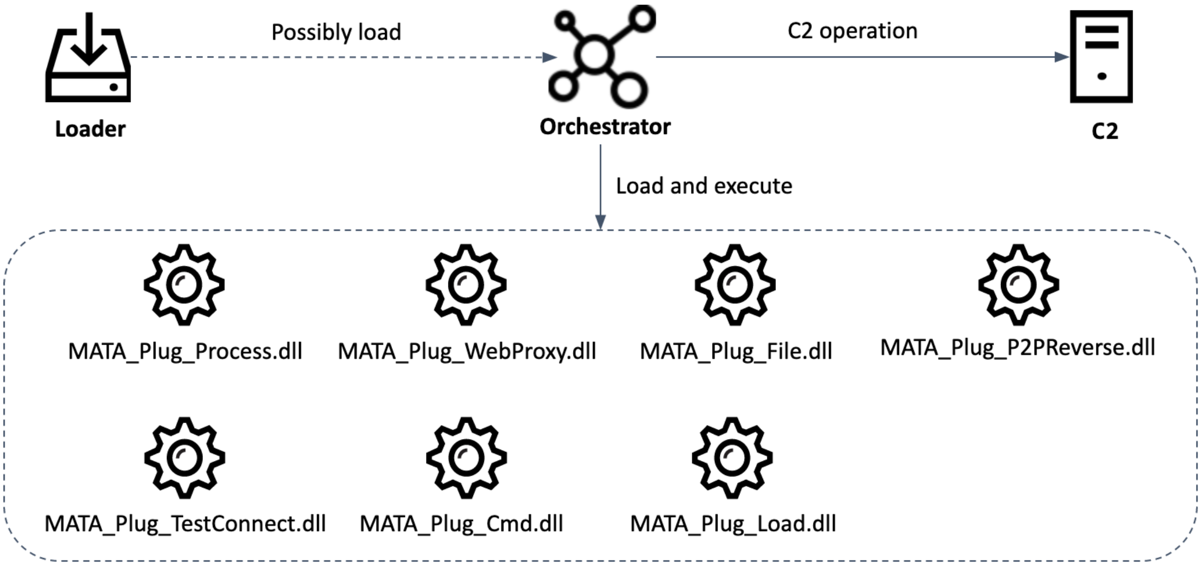
By GReAT

As the IT and OT environment becomes more complex, adversaries are quick to adapt their attack strategy. For example, as users' work environments diversify, adversaries are busy acquiring the TTPs to infiltrate systems. Recently, we reported to our Threat Intelligence Portal customers a similar malware framework that internally we called MATA. The MATA malware framework possesses several components, such as loader, orchestrator and plugins. This comprehensive framework is able to target Windows, Linux and macOS operating systems.

The first artefacts we found relating to MATA were used around April 2018. After that, the actor behind this advanced malware framework used it aggressively to infiltrate corporate entities around the world. We identified several victims from our telemetry and figured out the purpose of this malware framework.

Windows version of MATA

The Windows version of MATA consists of several components. According to our telemetry, the actor used a loader malware to load the encrypted next-stage payload. We're not sure that the loaded payload is the orchestrator malware, but almost all victims have the loader and orchestrator on the same machine.



Component of the Windows version of MATA

Loader

This loader takes a hardcoded hex-string, converts it to binary and AES-decrypts it in order to obtain the path to the payload file. Each loader has a hard-coded path to load the encrypted payload. The payload file is then AES-decrypted and loaded.

From the loader malware found on one of the compromised victims, we discovered that the parent process which executes the loader malware is the "C:\Windows\System32\wbem\WmiPrvSE.exe" process. The WmiPrvSE.exe process is "WMI Provider Host process", and it usually means the actor has executed this loader malware from a remote host to move laterally. Therefore, we assess that the actor used this loader to compromise additional hosts in the same network.

Orchestrator and plugins

We discovered the orchestrator malware in the lsass.exe process on victims' machines. This orchestrator malware loads encrypted configuration data from a registry key and decrypts it with the AES algorithm. Unless the registry value exists, the malware uses hard-coded configuration data. The following is a configuration value example from one orchestrator malware sample:

Victim ID	Random 24-bit number
Internal version number	3.1.1 (0x030101)
Timeout	20 minutes
C2 addresses	108.170.31[.]81:443 192.210.239[.]122:443 111.90.146[.]105:443
Disk path or URL of plugin (up to 15) to be loaded on start	Not used in this malware

The orchestrator can load 15 plugins at the same time. There are three ways to load them:

- Download the plugin from the specified HTTP or HTTPS server
- Load the AES-encrypted plugin file from a specified disk path
- Download the plugin file from the current MataNet connection

The malware authors call their infrastructure MataNet. For covert communication, they employ TLS1.2 connections with the help of the "openssl-1.1.0f" open source library, which is statically linked inside this module. Additionally, the traffic between MataNet nodes is encrypted with a random RC4 session key. MataNet implements both client and server mode. In server mode the certificate file "c_2910.cls" and the private key file "k_3872.cls" are loaded for TLS encryption. However, this mode is never used.

The MataNet client establishes periodic connections with its C2. Every message has a 12-byte-long header, where the first DWORD is the message ID and the rest is the auxiliary data, as described in the table below:

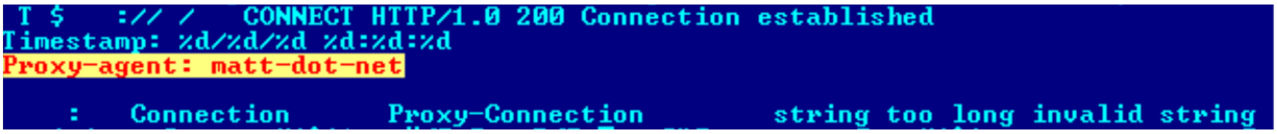
Message ID	Description
0x400	Complete the current MataNet session and delay the next session until the number of logical drives is changed or a new active user session is started.
0x500	Delete configuration registry key and stop MATA execution until next reboot.
0x601	Send configuration data to C2.
0x602	Download and set new configuration data.
0x700	Send the C2 the infected host basic information such as victim ID, internal version number, Windows version, computer name, user name, IP address and MAC address.
0x701	Send the C2 the configuration settings such as victim ID, internal version number and session timeout.

The main functionality of the orchestrator is loading each plugin file and executing them in memory. Each DLL file type plugin provides an interface for the orchestrator and provides rich functionality that can control infected machines.

Plugin name	Description
MATA_Plug_Cmd.dll	Run “cmd.exe /c” or “powershell.exe” with the specified parameters, and receive the output of the command execution.
MATA_Plug_Process.dll	Manipulate process (listing process, killing process, creating process, creating process with logged-on user session ID).
MATA_Plug_TestConnect.dll	Check TCP connection with given IP:port or IP range. Ping given host or IP range.
MATA_Plug_WebProxy.dll	Create a HTTP proxy server. The server listens for incoming TCP connections on the specified port, processing CONNECT requests from clients to the HTTP server and forwarding all traffic between client and server.
MATA_Plug_File.dll	Manipulate files (write received data to given file, send given file after LZNT1 compression, compress given folder to %TEMP%\~DESKTOP[8random hex].ZIP and send, wipe given file, search file, list file and folder, timestomping file).

MATA_Plug_Load.dll	Inject DLL file into the given process using PID and process name, or inject XORed DLL file into given process, optionally call export function with arguments.
MATA_Plug_P2PReverse.dll	Connect between MataNet server on one side and an arbitrary TCP server on the other, then forward traffic between them. IPs and ports for both sides are specified on the call to this interface.

There is an interesting string inside the MATA_Plug_WebProxy plugin – "Proxy-agent: *matt-dot-net*" – which is a reference to Matt McKnight's open source project. There are some differences though. Matt's project is written in C# rather than C++. The MATA proxy is noticeably simpler, as there is no cache and no SSL support, for instance. It's possible that MATA's authors found and used the source code of an early version of Matt's proxy server. It looks like the malware author rewrote the code from C# to C++ but left this footprint unchanged.



```
 T $   :// /   CONNECT HTTP/1.0 200 Connection established
Timestamp: %d/%d/%d %d:%d:%d
Proxy-agent: matt-dot-net

    :   Connection       Proxy-Connection       string too long invalid string
```

Proxy-agent of MATA_Plug_WebProxy.dll plugin

Non-Windows version of MATA

The MATA framework targets not only the Windows system but also Linux and macOS systems.

Linux version

During our research, we also found a package containing different MATA files together with a set of hacking tools. In this case, the package was found on a legitimate distribution site, which might indicate that this is the way the malware was distributed. It included a Windows MATA orchestrator, a Linux tool for listing folders, scripts for exploiting Atlassian Confluence Server (CVE-2019-3396), a legitimate socat tool and a Linux version of the MATA orchestrator bundled together with a set of plugins. China-based security vendor Netlab also published a highly detailed blog on this malware.

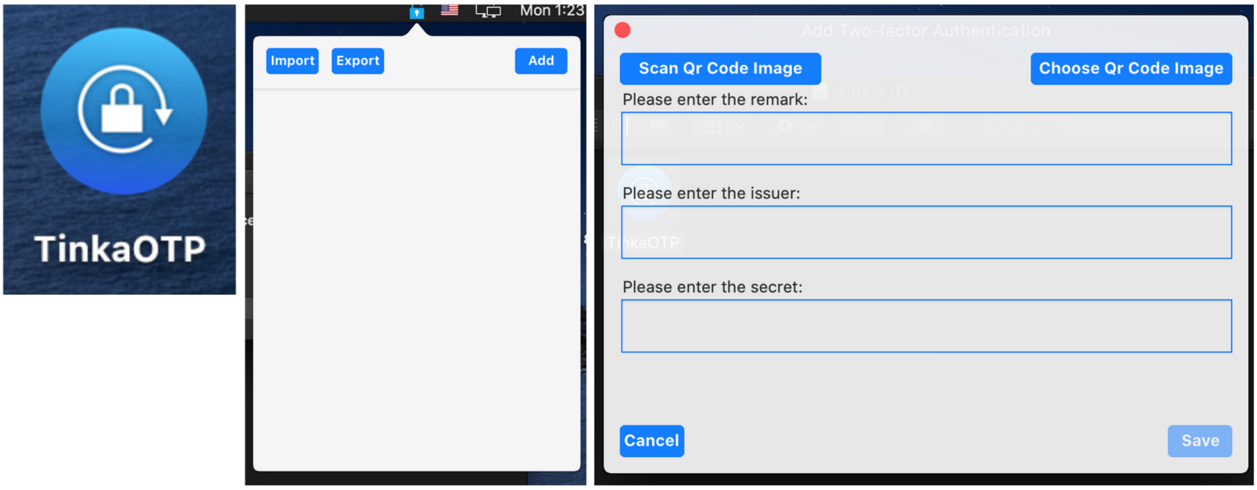
The module is designed to run as a daemon. Upon launch, the module checks if it is already running by reading the PID from "/var/run/init.pid" and checks if the "/proc/%pid%/cmdline" file content is equal to "/flash/bin/mountd". Note that "/flash/bin/mountd" is an unusual path for standard Linux desktop or server installations. This path suggests that MATA's Linux targets are diskless network devices such as routers, firewalls or IoT devices based on x86_64. The module can be run with the "/pro" switch to skip the "init.pid" check. The AES-encrypted configuration is stored in the “$HOME/.memcache” file. The behavior of this module is the same as the Windows MATA orchestrator previously described. The plugin names of Linux MATA and the corresponding Windows plugins are:

Linux plugin	Corresponding Windows plugin
/bin/bash	MATA_Plug_Cmd
plugin_file	MATA_Plug_File
plugin_process	MATA_Plug_Process
plugin_test	MATA_Plug_TestConnect
plugin_reverse_p2p	MATA_Plug_P2PReverse

Note that the Linux version of MATA has a *logsend* plugin. This plugin implements an interesting new feature, a “scan” command that tries to establish a TCP connection on ports 8291 (used for administration of MikroTik RouterOS devices) and 8292 (“Bloomberg Professional” software) and random IP addresses excluding addresses belonging to private networks. Any successful connection is logged and sent to the C2. These logs might be used by attackers for target selection.

macOS version

We discovered another MATA malware target for macOS uploaded to VirusTotal on April 8, 2020. The malicious Apple Disk Image file is a Trojanized macOS application based on an open-source two-factor authentication application named MinaOTP.



Please enter the remark:

Please enter the issuer:

Please enter the secret:

Trojanized macOS application

The Trojanized main TinkaOTP module is responsible for moving the malicious Mach-O file to the Library folder and executing it using the following command:
cp TinkaOTP.app/Contents/Resources/Base.lproj/**SubMenu.nib** ~/Library/.mina >

/dev/null 2>&1 && chmod +x ~/Library/.mina > /dev/null 2>&1 && ~/Library/.mina > /dev/null 2>&1

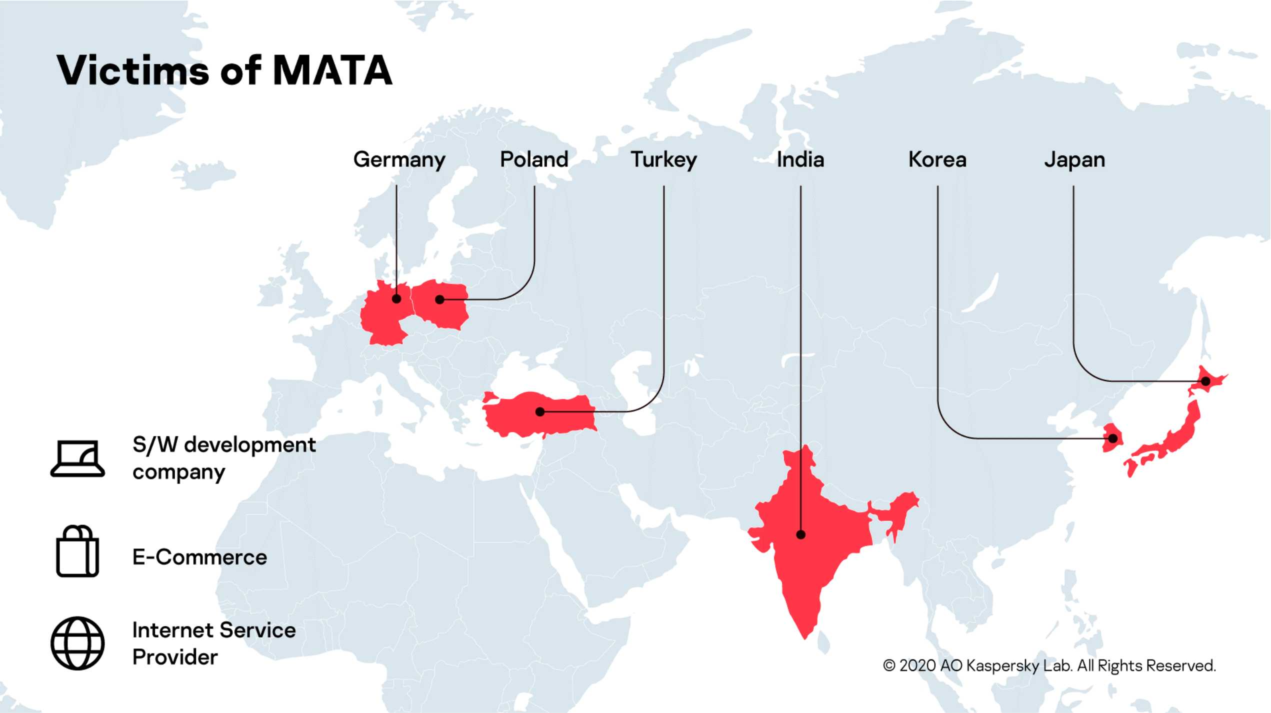
Upon launch, this malicious Mach-o file loads the initial configuration file from "/Library/Caches/com.apple.appstotore.db".

Like another strain running on a different platform, the macOS MATA malware also runs on a plugin basis. Its plugin list is almost identical to the Linux version, except that it also contains a plugin named "plugin_socks". The "plugin_socks" plugin is similar to "plugin_reverse_p2p" and is responsible for configuring proxy servers.

Victims

Based on our telemetry, we have been able to identify several victims who were infected by the MATA framework. The infection is not restricted to a specific territory. Victims were recorded in Poland, Germany, Turkey, Korea, Japan and India. Moreover, the actor compromised systems in various industries, including a software development company, an e-commerce company and an internet service provider.

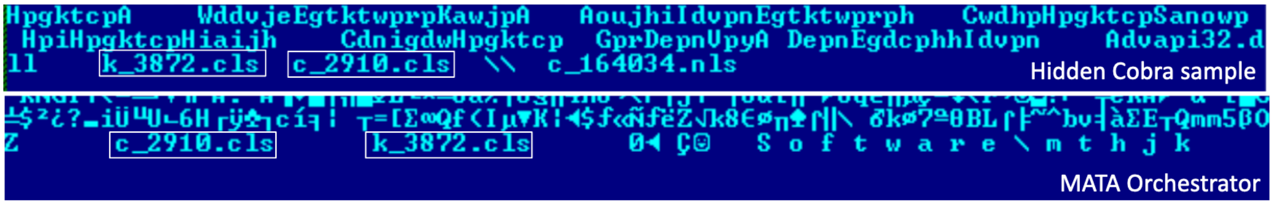
We assess that MATA was used by an APT actor, and from one victim we identified one of their intentions. After deploying MATA malware and its plugins, the actor attempted to find the victim's databases and execute several database queries to acquire customer lists. We're not sure if they completed the exfiltration of the customer database, but it's certain that customer databases from victims are one of their interests. In addition, MATA was used to distribute VHD ransomware to one victim, something that will be described in detail in an upcoming blog post.



Victims of MATA

Attribution

We assess that the MATA framework is linked to the Lazarus APT group. The MATA orchestrator uses two unique filenames, c_2910.cls and k_3872.cls, which have only previously been seen in several Manuscrypt variants, including the samples (0137f688436c468d43b3e50878ec1a1f) mentioned in the US-CERT publication.

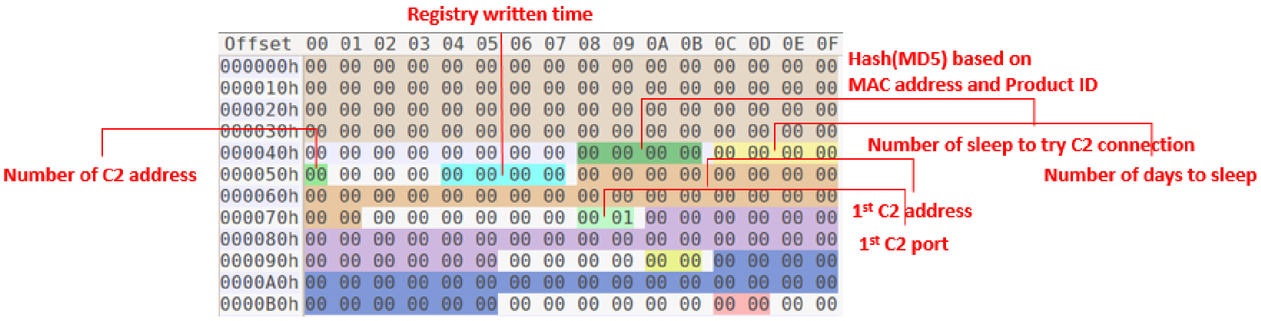


```
HpgktcpA     WddvjeEgtktwprpKawjpA    AoujhiIdvpnEgtktwprph    CwdhpHpgktcpSanowp
 HpiHpgktcpHiaijh     CdnigdwHpgktcp   GprDepnVpyA DepnEgdcphhIdvpn      Advapi32.d
ll      k_3872.cls  c_2910.cls   \\   c_164034.nls
```

```
≝$²¿?▁iÜ╙U╘6H╓ÿ╧╕cí╕¦ ┬=[Σ∞Qf<Iµ▼K¦◄$ƒ«ÑƒëZ√k8ϵø╥♠╓║\ δkø7ºθBL╒╞~^bv╡àΣE╥Qmm5ßO
Z       c_2910.cls        k_3872.cls        0◄ Ç☺   S o f t w a r e \ m t h j k
```

Unique file name

Moreover, MATA uses global configuration data including a randomly generated session ID, date-based version information, a sleep interval and multiple C2s and C2 server addresses. We've seen that one of the Manuscrypt variants (ab09f6a249ca88d1a036eee7a02cdd16) shares a similar configuration structure with the MATA framework. This old Manuscrypt variant is an active backdoor that has similar configuration data such as session ID, sleep interval, number of C2 addresses, infected date, and C2 addresses. They are not identical, but they have a similar structure.



```
Offset  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000010h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000020h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000030h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000040h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000050h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000060h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000070h 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00
000080h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000090h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000A0h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000B0h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Manuscrypt configuration structure

Conclusion

The MATA framework is significant in that it is able to target multiple platforms: Windows, Linux and macOS. In addition, the actor behind this advanced malware framework utilized it for a type of cybercrime attack that steals customer databases and distributes ransomware. We evaluate that this malware is going to evolve, so we will be monitoring its activity in order to protect our customers.

For more information please contact: intelreports@kaspersky.com

Indicators of compromise

File Hashes (malicious documents, Trojans, emails, decoys)

Windows Loader

f364b46d8aafff67271d350b8271505a
85dcea03016df4880cebee9a70de0c02
1060702fe4e670eda8c0433c5966feee
7b068dfbea310962361abf4723332b3a
8e665562b9e187585a3f32923cc1f889
6cd06403f36ad20a3492060c9dc14d80
71d8b4c4411f7ffa89919a3251e6e5cb
a7bda9b5c579254114fab05ec751918c
e58cfbc6e0602681ff1841afadad4cc6
7e4e49d74b59cc9cc1471e33e50475d3
a93d1d5c2cb9c728fda3a5beaf0a0ffc
455997E42E20C8256A494FA5556F7333
7ead1fbba01a76467d63c4a216cf2902
7d80175ea344b1c849ead7ca5a82ac94
bf2765175d6fce7069cdb164603bd7dc
b5d85cfaece7da5ed20d8eb2c9fa477c
6145fa69a6e42a0bf6a8f7c12005636b
2b8ff2a971555390b37f75cb07ae84bd
1e175231206cd7f80de4f6d86399c079
65632998063ff116417b04b65fdebdfb
ab2a98d3564c6bf656b8347681ecc2be
e3dee2d65512b99a362a1dbf6726ba9c
fea3a39f97c00a6c8a589ff48bcc5a8c
2cd1f7f17153880fd80eba65b827d344
582b9801698c0c1614dbbae73c409efb
a64b3278cc8f8b75e3c86b6a1faa6686
ca250f3c7a3098964a89d879333ac7c8
ed5458de272171feee479c355ab4a9f3
f0e87707fd0462162e1aecb6b4a53a89
f1ca9c730c8b5169fe095d385bac77e7
f50a0cd229b7bf57fcbd67ccfa8a5147

Windows MATA

bea49839390e4f1eb3cb38d0fcaf897e rdata.dat
8910bdaaa6d3d40e9f60523d3a34f914 sdata.dat
6a066cf853fe51e3398ef773d016a4a8
228998f29864603fd4966cadd0be77fc
da50a7a05abffb806f4a60c461521f41
ec05817e19039c2f6cc2c021e2ea0016

Registry path

HKLM\Software\Microsoft\KxtNet
HKLM\Software\Microsoft\HlqNet
HKLM\Software\mthjk

Linux MATA

859e7e9a11b37d355955f85b9a305fec	mdata.dat
80c0efb9e129f7f9b05a783df6959812	ldata.dat, mdata.dat
d2f94e178c254669fb9656d5513356d2	mdata.dat

Linux log collector

982bf527b9fe16205fea606d1beed7fa	hdata.dat

Open-source Linux SoCat

e883bf5fd22eb6237eb84d80bbcf2ac9	sdata.dat

Script for exploiting Atlassian Confluence Server

a99b7ef095f44cf35453465c64f0c70c	check.vm, r.vm
199b4c116ac14964e9646b2f27595156	r.vm

macOS MATA

81f8f0526740b55fe484c42126cd8396	TinkaOTP.dmg
f05437d510287448325bac98a1378de1	SubMenu.nib

C2 address

104.232.71.7:443
107.172.197.175:443
108.170.31.81:443
111.90.146.105:443
111.90.148.132:443
172.81.132.41:443
172.93.184.62:443
172.93.201.219:443
185.62.58.207:443
192.210.239.122:443
198.180.198.6:443
209.90.234.34:443
216.244.71.233:443
23.227.199.53:443
23.227.199.69:443

23.254.119.12:443
67.43.239.146:443
68.168.123.86:443