Sandman APT | China-Based Adversaries Embrace Lua

Aleksandar Milenkoski ⋮

By Aleksandar Milenkoski, Bendik Hagen (PwC), and Microsoft Threat Intelligence

Executive Summary

- The Sandman APT is likely associated with suspected China-based threat clusters known to use the KEYPLUG backdoor, in particular a cluster jointly presented by PwC and Microsoft at Labscon 2023 – STORM-0866/Red Dev 40.
- The Sandman's Lua-based malware LuaDream and the KEYPLUG backdoor were observed co-existing in the same victim environments.
- Sandman and STORM-0866/Red Dev 40 share infrastructure control and management practices, including hosting provider selections, and domain naming conventions.
- The implementation of LuaDream and KEYPLUG reveals indicators of shared development practices and overlaps in functionalities and design, suggesting shared functional requirements by their operators.
- The use of the Lua development paradigm in the cyberespionage domain, historically associated with actors considered Western or Western-aligned, is likely being adopted by a broader range of adversaries, including those with ties to China.

Overview

In this report, SentinelLabs, Microsoft, and PwC threat intelligence researchers provide attribution-relevant information on the Sandman APT cluster positioning this threat on the broader threat landscape. We highlight links between Sandman and a suspected China-based threat actor using the shared KEYPLUG backdoor – STORM-0866/Red Dev 40. This includes victimology overlaps, cohabitation, and sharing C2 infrastructure control and management practices.

STORM-0866/Red Dev 40 is a developing APT threat cluster primarily targeting entities in the Middle East and the South Asian subcontinent, including telecommunication providers and government entities. These are regions and sectors where we also observed Sandman activity. The modular backdoor KEYPLUG is a staple in STORM-0866/Red Dev 40's arsenal. Mandiant first reported on KEYPLUG as part of intrusions into U.S. government entities by the Chinese APT group APT41.

Microsoft and PwC have subsequently identified at least three other developing clusters involving KEYPLUG, including STORM-0866/Red Dev 40. Their research, making the case that KEYPLUG is likely shared among multiple suspected China-based groups, was presented at LabsCon 2023. They distinguish STORM-0866/Red Dev 40 from the other clusters based on specific malware characteristics, such as unique encryption keys for KEYPLUG C2 communication, and a higher sense of operational security, such as relying on Cloud-based reverse proxy infrastructure for hiding the true hosting locations of their C2 servers.

SentinelLabs and Microsoft have observed Sandman's LuaDream and KEYPLUG implants cohabiting in the same victim environments, some of which are on the same endpoints. LuaDream is a maintained modular backdoor based on LuaJIT, with version 11.0.2.1.23.1 observed in March 2023 and version 12.0.2.5.23.29 observed in August 2023. In one instance, the KEYPLUG malware had been deployed approximately 3 months prior to LuaDream (in May 2023). LuaDream and KEYPLUG were active at the same time over approximately 2 weeks until both threats were remediated. During this time period, we did not observe any contestation or deconfliction activities by the LuaDream or KEYPLUG operators.

A close examination of the implementation and C2 infrastructure of these distinct malware strains revealed indicators of shared development as well as infrastructure control and management practices, and some overlaps in functionalities and design, suggesting shared functional requirements by their operators.

The findings we present are yet another showcase of the complex nature of the China-based threat landscape. As exemplified by Sandman and STORM-0866/Red Dev 40, this landscape is marked by substantial cooperation and coordination among its constituent threat groups, along with the possibility of third-party vendors supplying the operational teams with tooling. This makes accurate clustering challenging. Therefore, while acknowledging the association of Sandman with the suspected China-based adversaries using KEYPLUG, we continue to track Sandman as a distinct cluster until further conclusive information suggesting otherwise becomes available.

Lua-based modular backdoors, such as LuaDream, have been observed relatively rarely and often in the context of espionage-motivated APTs historically considered Western or Western-aligned. Our findings on Sandman indicate that the Lua development paradigm is being adopted by a broader set of cyberespionage threat actors for the modularity, portability, and simplicity that the Lua scripting language offers.

Sandman and STORM-0866/Red Dev 40 Infrastructure

The SSL certificate assigned to the LuaDream C2 domain `ssl.explorecell[.]com` has also been used on the servers with IPs of `185.51.134[.]27` (between March and April 2023) and `45.80.148[.]151` (in March 2023). `185.51.134[.]27` is allocated to the Estonian VPS service provider EstNOC and `45.80.148[.]151` to the Romanian provider HOSTGW SRL. `ssl.explorecell[.]com` last resolved to `185.82.218[.]230`, an IP address of a server hosted in Bulgaria by the ITLDC hosting provider.

- Thumbprint: fc8fdf58cd945619cbfede40ba06aada10de9459
- Serial number: 364670096077097330220756280372394037039639
- Common Name: `ssl.explorecell[.]com`

Approximately 4 months later (in August 2023), the server at `185.51.134[.]27` used an SSL certificate issued for the domain `dan.det-ploshadka[.]com`. This domain last resolved to 79.110.52[.]160, a server hosted by the Romanian service provider M247.

- Thumbprint: a7932112b7880c95d77bc36c6fcced977f4a5889
- Serial number: 365025056055127017786055050446086862849019
- Common Name: `dan.det-ploshadka[.]com`

Microsoft and PwC have observed `dan.det-ploshadka[.]com` being used as a KEYPLUG C2 server and attribute the domain with high confidence to STORM-0866/Red Dev 40. This assessment is primarily based on the use of RC4 keys for encrypting C2 data that are unique to STORM-0866/Red Dev 40 as well as used known STORM-0866/Red Dev 40 malware in the intrusions.

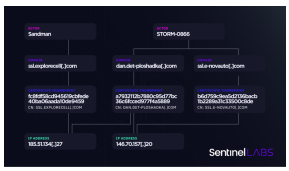
The `dan.det-ploshadka[.]com` certificate has also been used on the servers with IPs 45.90.59[.]17 (between July and September 2023), `45.129.199[.]122` (in September 2023), and `146.70.157[.]20` (in June 2023).

Another certificate, issued for the domain `ssl.e-novauto[.]com`, was also used on `146.70.157[.]20` in May 2023. `ssl.e-novauto[.]com`, which has an overlap in subdomain naming convention with the `ssl.explorecell[.]com` Sandman domain, last resolved to `172.67.216[.]63` (an IP address of a Cloud-based reverse proxy infrastructure). `146.70.157[.]20` is allocated to the Romanian hosting service provider M247.

- Thumbprint: b6d759c9ea5d2136bacb1b2289a31c33500c8de8
- Serial number: 599612378987262804627462177924300244018152830 68
- Common Name: `ssl.e-novauto[.]com`

In common with `dan.det-ploshadka[.]com`, Microsoft and PwC have observed the `ssl.e-novauto[.]com` domain being used as a KEYPLUG C2 server and attribute the domain with high confidence to STORM-0866/Red Dev 40.

Among the other server IPs on which the `ssl.e-novauto[.]com` certificate was used (`5.255.88[.]188` in October 2022; `5.2.67[.]176` between March and May 2023; `5.2.72[.]130` in April 2022; `37.120.140[.]205` between March 2022 and May 2023; and 185.38.142[.]129 between October 2022 and January 2023), `5.2.67[.]176` has been the resolving IP for the `ssl.articella[.]com` domain since January 2023. This domain has an overlap in naming convention with the `ssl.e-novauto[.]com` STORM-0866/Red Dev 40 domain and the `ssl.explorecell[.]com` Sandman domain.



Infrastructure overview

PwC tracks STORM-0866/Red Dev 40 as a distinct cluster from the other threat groups using the KEYPLUG malware based on their frequent use of Cloud-based reverse proxy infrastructure, likely as an operational security measure to avoid exposing the true hosting locations. We observed this in the context of Sandman as well, noting a shift from

using a directly exposed C2 server IP address (C2 domain: `ssl.explorecell[.]com`) to address of a reverse proxy infrastructure (C2 domain: `mode.encagil[.]com`).

The overlap of unique infrastructure control and management practices, hosting provider selections, and domain naming conventions, indicates a likely relation between the Sandman and the STORM-0866/Red Dev 40 APT clusters from an infrastructure perspective.

LuaDream and KEYPLUG

LuaDream and KEYPLUG are distinct malware strains. KEYPLUG is implemented in C++, whereas the majority of the LuaDream functionalities are implemented in Lua. The samples that we analyzed do not share straightforward indicators that would confidently classify them as closely related or originating from the same source, such as use of identical encryption keys or direct overlaps in implementation. However, we observed indicators of shared development practices and some overlaps in functionalities and design, suggesting shared functional requirements by the operators. This is not uncommon in the Chinese malware landscape.

We also observed a code comment in Chinese in the `main_proto_WinHttpServer` component of LuaDream version 11.0.2.1.23.1, indicating potential Chinese origin. However, we note that all other LuaDream string artifacts (function and variable names, and code comment, status, and error reporting strings) are formulated in English.

```
//返回的句柄;
typedef struct _WIN_HTTP_SERVER_HANDLE_
{
    HANDLE          hReqQueue;
    char szTokenKey[MAX_PATH];
    char szDataKey[MAX_PATH];
    [...]
```

Code comment in LuaDream (translates from Chinese to “returned handle”)

LuaDream is likely still in active development. It remains to be seen whether further iterations of the malware and its plugins will share implementation overlaps, functionality or design patterns with KEYPLUG or other malware strains of suspected Chinese origin.

C2 Protocols

LuaDream and KEYPLUG are highly modular and multi-protocol in design, both implementing support for the HTTP, TCP, WebSocket, and QUIC protocols for C2 communication. The combination of QUIC and WebSocket is a relatively rare backdoor feature and its implementation in both LuaDream and KEYPLUG may be the result of a shared functional requirement by the backdoors’ operators.

The order in which LuaDream and KEYPLUG evaluate the configured protocol among HTTP, TCP, WebSocket, and QUIC is the same: HTTP, TCP, WebSocket, and QUIC in that order. The LuaDream keyword `HTTPS2` refers to WebSocket and KEYPLUG implements additional support for UDP. We do not exclude the possibility for future versions of LuaDream to support UDP as well.

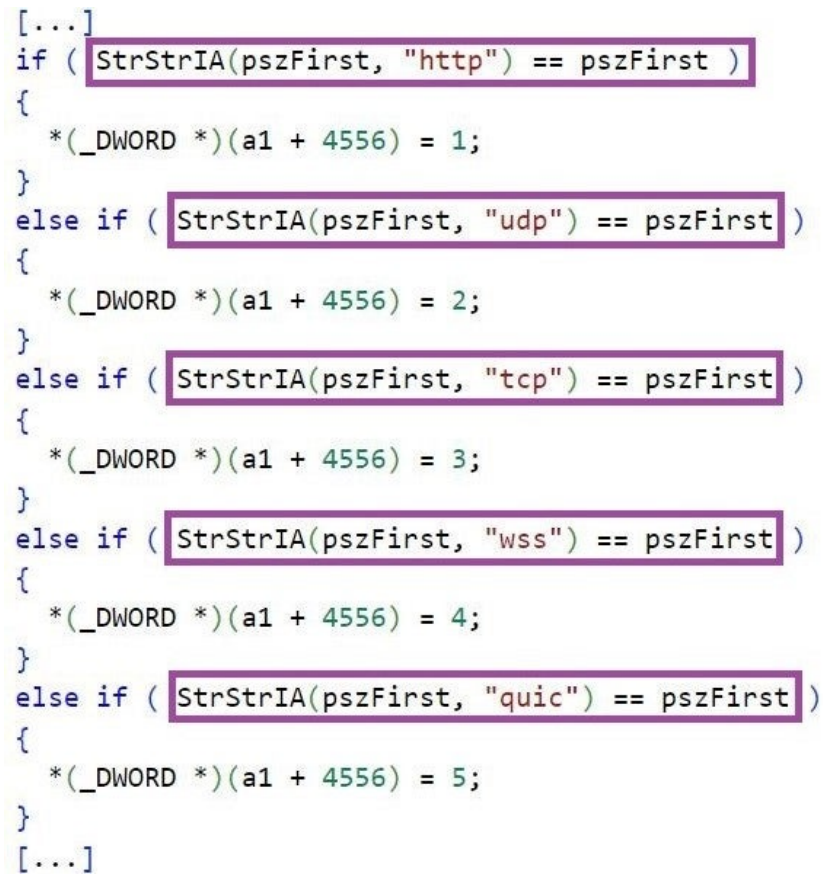
```
[...]
elseif uv0.string(gPm[0].ConnectInfo.protoType) == "HTTP" or
       uv0.string(gPm[0].ConnectInfo.protoType) == "HTTPS" then
    slot1, slot3, slot4 = winhttpclient_recv(uv0.cast("PWIN_HTTP_CLIENT_HANDLE_", slot0))

elseif uv0.string(gPm[0].ConnectInfo.protoType) == "TCP" then
    slot1, slot3, slot4 = tcpclient_recv(uv0.cast("PTCPCLIENT_HANDLE_", slot0))
    gPm[0].ConnectInfo.heartTime = uv1.GetTickCount64()

elseif uv0.string(gPm[0].ConnectInfo.protoType) == "HTTPS2" then
    slot1, slot3, slot4 = websocketclient_recv(uv0.cast("PWEBCLIENT_HANDLE_", slot0))
    gPm[0].ConnectInfo.heartTime = uv1.GetTickCount64()

elseif uv0.string(gPm[0].ConnectInfo.protoType) == "QUIC" then
    slot1, slot3, slot4 = quic_recv(uv0.cast("PQUICCLIENT_HANDLE_", slot0))
    gPm[0].ConnectInfo.heartTime = uv1.GetTickCount64()
[...]
```

LuaDream: Protocol handling



KEYPLUG: Protocol handling

For each protocol, both LuaDream and KEYPLUG implement internal structures that store client data, such as the handles to the established sockets to the C2 servers.

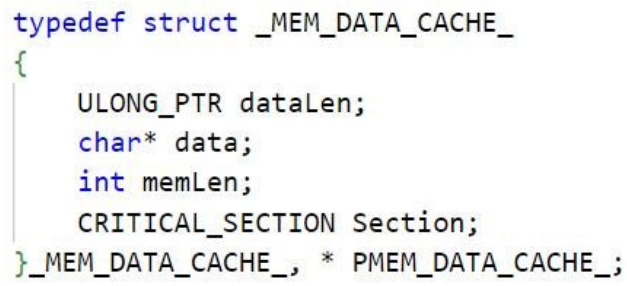
Execution Flow and C2 Data Management

The high-level execution flows of LuaDream and KEYPLUG are very similar. Both backdoors first gather and exfiltrate system and user information in designated functions, with overlaps in gathered information (for example, MAC address, OS version, IP address, computer name, and username).

LuaDream and KEYPLUG then instantiate threads designated for sending and receiving C2 data, establish connection to the C2 server, and continue to process backdoor commands and manage plugins. Plugin management includes loading and unloading plugins.

The backdoors use global data buffers designated for storing data to be sent to the C2 server, and data received from the server. LuaDream and KEYPLUG read from the global buffers that store incoming C2 data and continue processing it when available.

LuaDream and KEYPLUG store in designated internal structures overlapping information about the global buffers, such as starting memory addresses, sizes, and pointers to Windows `CRITICAL_SECTION` structures. LuaDream defines this structure as `_MEM_DATA_CACHE_`.



LuaDream: Global buffer structure (decompiled LuaJIT bytecode)

```
struct _MEM_RECV_CACHE
{
  _QWORD BufferR_vtable;
  [...]
  _QWORD data;
  _QWORD memLen;
  [...]
  CRITICAL_SECTION CriticalSection1;
  CRITICAL_SECTION CriticalSection2;
  [...]
};
```

KEYPLUG: Global buffer structure (IDA-defined structure)

LuaDream and KEYPLUG implement designated functions for reading from, and writing to, these buffers. These functions synchronize buffer access by multiple threads using Windows Critical Sections.

```
__int64 __fastcall GetDataCache(_MEM_DATA_CACHE_ *pRecvCache,
                                void *dest, int *a3)
{
  size_t v6;
  const void *data;

  if ( !pRecvCache || !dest || !*a3 )
    return 0i64;
  RtlEnterCriticalSection(&pRecvCache->Section);
  v6 = *a3;
  data = (const void *)pRecvCache->data;
  if ( pRecvCache->dataLen < v6 )
  {
    memmove(dest, data, pRecvCache->dataLen);
    memset((void *)pRecvCache->data, 0, pRecvCache->dataLen);
  [...]
  RtlLeaveCriticalSection(&pRecvCache->Section);
  return 1i64;
}
```

LuaDream: Reading C2 data from a global buffer

```
__int64 __fastcall GetDataCache(_MEM_RECV_CACHE *pRecvCache,
                                char *dest, signed int a3)
{
  struct _RTL_CRITICAL_SECTION *p_CriticalSection;
  [...]
  p_CriticalSection = (struct _RTL_CRITICAL_SECTION *)&pRecvCache->CriticalSection1;
  if ( pRecvCache != (_MEM_RECV_CACHE *)0xFFFFFFFFFFFFFFD8i64 )
    EnterCriticalSection((LPCRITICAL_SECTION)&pRecvCache->CriticalSection1);
  [...]
  v1 = SHIDWORD(pRecvCache->BufferSize);
  [...]
  Buffer = pRecvCache->Buffer;
  [...]
    memmove(dest, v11, v12);
    v13 = (const void *)pRecvCache->Buffer;
    [...]
    HIDWORD(pRecvCache->BufferSize) = v14;
    memmove(&dest[v12], v13, v14);
  [...]
  if ( p_CriticalSection )
    LeaveCriticalSection(p_CriticalSection);
  return (unsigned int)a3;
}
```

KEYPLUG: Reading C2 data from a global buffer

Throughout their execution, both LuaDream and KEYPLUG generate one-time integer values based on the system uptime returned by the GetTickCount function. The backdoors calculate these values by applying modulo and/or addition operations to the system uptime. Some overlapping uses of the generated values are as sleep time intervals or protocol-specific keys, such as the `Sec-WebSocket-Key` packet header field that is used in the WebSocket opening handshake.

```
[...]
TickCount = GetTickCount();
sleep(TickCount % 0x1247);
[...]
```

LuaDream: Sleep interval

```
[...]
TickCount = GetTickCount();
Sleep(TickCount % 0x3A980 + 240000);
[...]
```

KEYPLUG: Sleep interval

Conclusions

We assess that there are strong overlaps in operational infrastructure, targeting, and TTPs associating the Sandman APT with China-based adversaries using the KEYPLUG backdoor, STORM-0866/Red Dev 40 in particular. This highlights the complex nature of the Chinese threat landscape. Its constituent threat actors will almost certainly continue to cooperate and coordinate, exploring new approaches to upgrade the functionality, flexibility, and stealthiness of their malware. The adoption of the Lua development paradigm is a compelling illustration of this.

Navigating the threat landscape calls for continuous collaboration and information sharing within the threat intelligence research community. SentinelLabs remains committed to this mission and is grateful to our industry partners involved in this collective endeavor.

Indicators of Compromise

Domains

dan.det-ploshadka[.]com	KEYPLUG C2 server
mode.encagil[.]com	LuaDream C2 server
ssl.articella[.]com	Suspected KEYPLUG or LuaDream C2 server

ssl.e-novauto[.]com	KEYPLUG C2 server
ssl.explorecell[.]com	LuaDream C2 server
yum.luxyries[.]com	KEYPLUG C2 server

IP Addresses

146.70.157[.]20	KEYPLUG C2 server (based on known C2 certificates)
172.67.216[.]63	KEYPLUG C2 server
185.38.142[.]129	KEYPLUG C2 server (based on a known C2 certificate)
185.51.134[.]27	LuaDream and KEYPLUG C2 (based on known C2 certificates)
185.82.218[.]230	LuaDream C2 server
37.120.140[.]205	KEYPLUG C2 server (according to a known C2 certificate)
45.129.199[.]122	KEYPLUG C2 server (based on a known C2 certificate)
45.80.148[.]151	LuaDream C2 (based on a known C2 certificate)
45.90.59[.]17	KEYPLUG C2 server (according to a known C2 certificate)
5.2.67[.]176	KEYPLUG C2 server (based on a known C2 certificate)
5.2.72[.]130	KEYPLUG C2 server (based on a known C2 certificate)
5.255.88[.]188	KEYPLUG C2 server (based on a known C2 certificate)
79.110.52[.]160	KEYPLUG C2 server

Certificate Thumbprints

a7932112b7880c95d77bc36c6fcced977f4a5889	KEYPLUG C2
b6d759c9ea5d2136bacb1b2289a31c33500c8de8	KEYPLUG C2
fc8fdf58cd945619cbfede40ba06aada10de9459	LuaDream C2