SANS ISC: InfoSec Handlers Diary Blog - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training InfoSec Handlers Diary Blog

isc.sans.edu/diary/Translating+Saitama's+DNS+tunneling+messages/28738

Translating Saitama's DNS tunneling messages

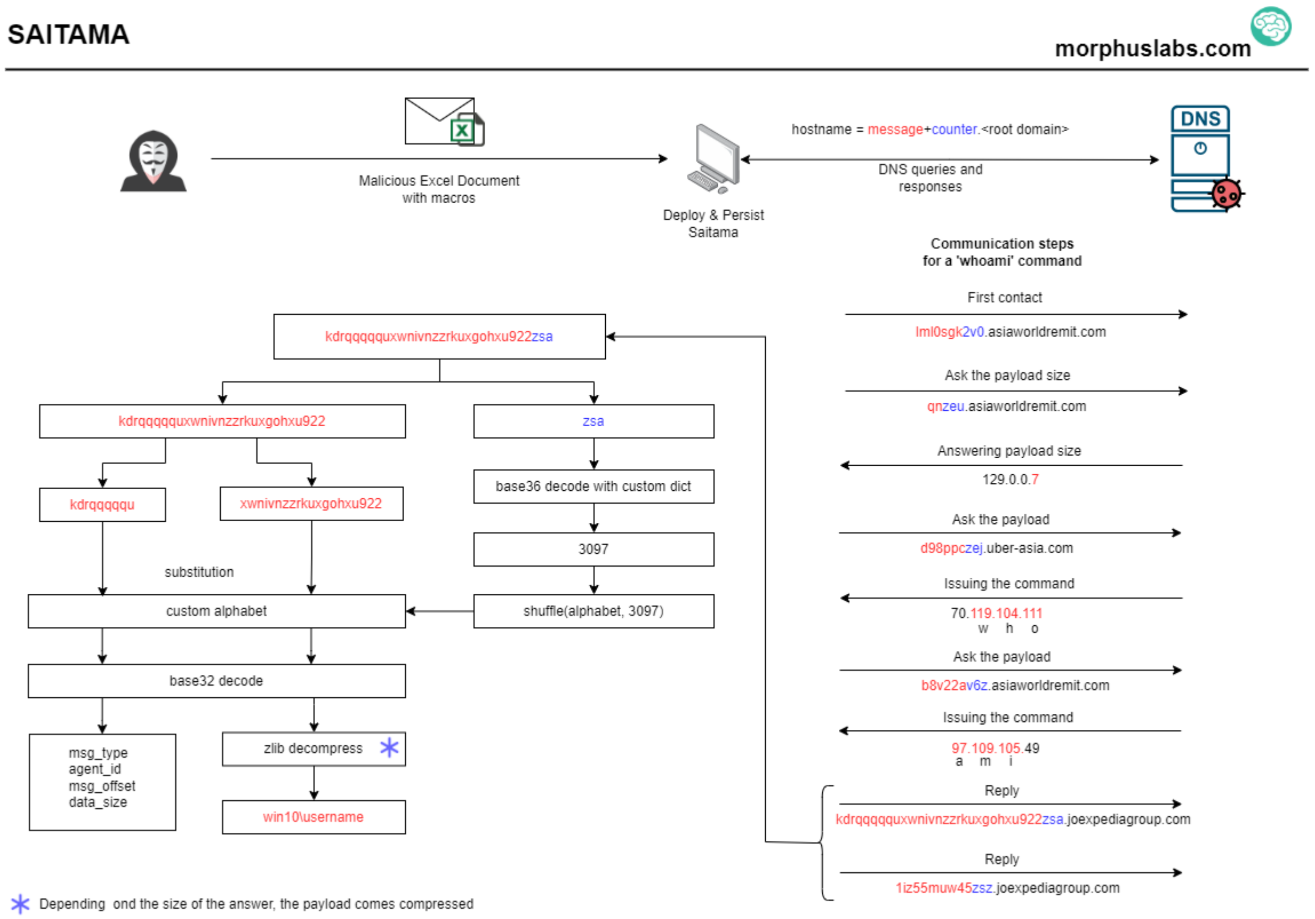
Published: 2022-06-13
Last Updated: 2022-06-13 15:00:45 UTC
by Renato Marinho (Version: 1)
0 comment(s)
Saitama is a **backdoor** that uses the DNS protocol to encapsulate its **command and control** (C2) messages - a technique known as DNS Tunneling (MITRE ATT&CK T1071). Spotted and documented by MalwareBytes in two articles posted last month (How the Saitama backdoor uses DNS tunneling and APT34 targets Jordan Government using new Saitama backdoor), Saitama was used in a phishing e-mail targeted to a government official from Jordan’s foreign ministry on an attack attributed to the Iranian group APT34.

Saitama caught my attention for two reasons: the stealth way the C2 messages are hidden in DNS protocol and the ease of access to malware implementation details by simply decompiling the .Net binary. Those points may increase the potential for other groups to use similar DNS tunneling techniques.

Saitama's DNS tunneling stealth strategy

Command and control (C2) over DNS is not new. It is common for a Victim <-> C2 communication occurs in the following way: data can be exfiltrated or answered from the victim to a C2 encoded in the hostname portion of the FQDN (i.e., oxn009lc7n5887k96c4zfckes6uif.rootdomain.com). In the other rand, commands or additional payloads can be downloaded from the C2 by the victim by querying TXT records to the attacker's controlled DNS server.

Saitama's implementation differs by not using TXT or other records able to store large data to encapsulate orders from the C2 to the victim. Instead, the orders are encapsulated in the IPV4 addresses themselves. For example, to issue the command '**whoami**', the server will answer two IP addresses: 70.**119.104.111** and **97.109.105**.49. The first octet (70) has a special meaning to the Saitama (a command will be issued), and the following octets are the ASCII code of the 'whoami' characters: **w=119, h=104, o=111**, and so on until **i=105**. The remaining octet is discarded. Look at the image below the communication between a victim and the C2 by issuing the command 'whoami':


SAITAMA
morphuslabs.com
Malicious Excel Document with macros
Deploy & Persist Saitama
hostname = message+counter.<root domain>
DNS queries and responses
DNS
Communication steps for a 'whoami' command
First contact
lml0sgk2v0.asiaworldremit.com
Ask the payload size
qnzeu.asiaworldremit.com
Answering payload size
129.0.0.7
Ask the payload
d98ppczej.uber-asia.com
Issuing the command
70.119.104.111
w h o
Ask the payload
b8v22av6z.asiaworldremit.com
Issuing the command
97.109.105.49
a m i
Reply
kdrqqqquxwnivnzzrkuxgohxu922zsa.joexpediagroup.com
Reply
1iz55muw45zsz.joexpediagroup.com
kdrqqqquxwnivnzzrkuxgohxu922zsa
kdrqqqquxwnivnzzrkuxgohxu922
zsa
kdrqqqqu
xwnivnzzrkuxgohxu922
base36 decode with custom dict
3097
substitution
custom alphabet
shuffle(alphabet, 3097)
base32 decode
msg_type
agent_id
msg_offset
data_size
zlib decompress
win10\username
Depending ond the size of the answer, the payload comes compressed

Saitama Translator

After analyzing Saitama's code, we developed a simple tool (available at https://github.com/morphuslabs/saitama_translator) capable of translating/decrypting the messages issued by the infected victim to the C2 server (the DNS queries). It may be helpful for those who face Saimanta or variants messages and need to try to discover what data is being sent to the C2.

Usage examples:

1. Passing one FQDN. In this case, the first response from Saitama do C2 after executing the command 'ver' on the infected system:

```
$ python translate.py vy5xxxxvzz650coacbsf03f2jkviwui9.joexpediagroup.com
agent_id: 114,  msg_type: 1,    msg_offset:0,   msg_size:43,
msg_content:b'9Microsoft W',
request:vy5xxxxvzz650coacbsf03f2jkviwui9.joexpediagroup.com,    count:4749
```

2. Passing multiple FQDN at once. In this case, all the responses from Saitama to C2 after executing 'ver' command:

```
$ python translate.py vy5xxxxvzz650coacbsf03f2jkviwui9.joexpediagroup.com
oxn009lc7n5887k96c4zfckes6uif.joexpediagroup.com pqxwwk9cyl1upnxwyqwinn0wgzui5.uber-
asia.com w7irwrisb5lxwkow81udr.uber-asia.com
agent_id: 114,  msg_type: 1,    msg_offset:0,   msg_size:43,
msg_content:b'9Microsoft W',
request:vy5xxxxvzz650coacbsf03f2jkviwui9.joexpediagroup.com,    count:4749
agent_id: 114,  msg_type: 1,    msg_offset:12,  msg_size:None,  msg_content:b'indows
[Vers',    request:oxn009lc7n5887k96c4zfckes6uif.joexpediagroup.com,      count:4750
agent_id: 114,  msg_type: 1,    msg_offset:24,  msg_size:None,  msg_content:b'ion
10.0.183',    request:pqxwwk9cyl1upnxwyqwinn0wgzui5.uber-asia.com,    count:4751
agent_id: 114,  msg_type: 1,    msg_offset:36,  msg_size:None,
msg_content:b'63.418]', request:w7irwrisb5lxwkow81udr.uber-asia.com,    count:4752
```

Notice that the string **"Microsoft Windows [Version 10.0.18363.418]"** was sent to the C2 server in four requests.

Saitama sample

e0872958b8d3824089e5e1cfab03d9d98d22b9bcb294463818d721380075a52d

References

https://attack.mitre.org/techniques/T1071/004

https://blog.malwarebytes.com/threat-intelligence/2022/05/how-the-saitama-backdoor-uses-dns-tunnelling/

https://blog.malwarebytes.com/threat-intelligence/2022/05/apt34-targets-jordan-government-using-new-saitama-backdoor/

Keywords:

0 comment(s)

DEV522 Defending Web Application Security Essentials
LEARN MORE
Learn to defend your apps before they're hacked

Top of page

×

Diary Archives