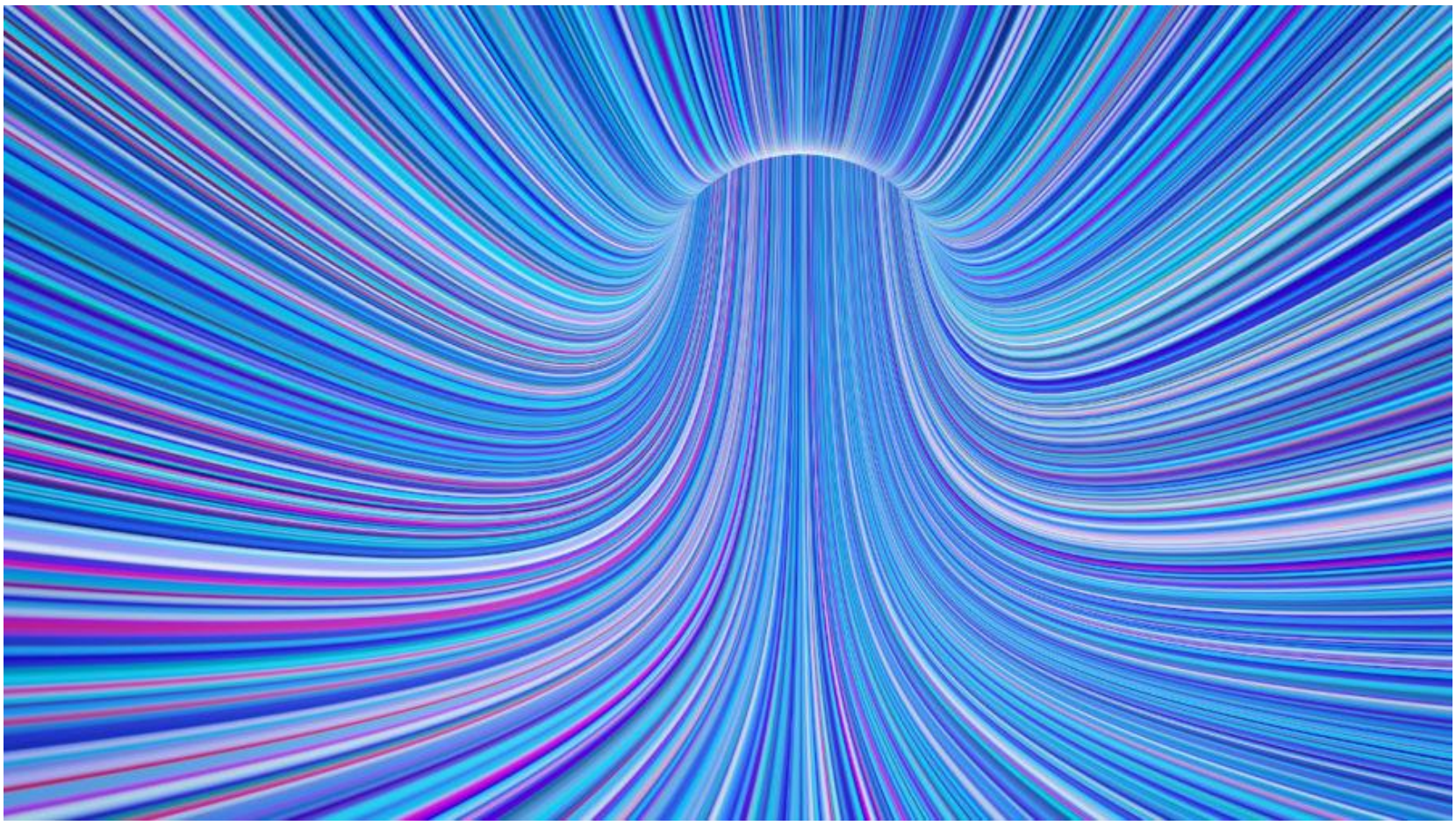

Explained | Threat Intelligence

How the Saitama backdoor uses DNS tunnelling

Posted: May 25, 2022 by Mark Stockley

A walkthrough of one of the stealthy communication techniques employed in a recent attack using APT34's Saitama backdoor.

Thanks to the Malwarebytes Threat Intelligence Team for the information they provided for this article.

Understandably, a lot of cybersecurity research and commentary focuses on the act of breaking into computers undetected. But threat actors are often just as concerned with the act of breaking out of computers undetected too.

Malware with the intent of surveillance or espionage needs to operate undetected, but the chances are it also needs to exfiltrate data or exchange messages with its command and control infrastructure, both of which could reveal its presence to threat hunters.

One of the stealthy communication techniques employed by malware trying to avoid detection is DNS Tunnelling, which hides messages inside ordinary-looking DNS requests.

The Malwarebytes Threat Intelligence team recently published research about an attack on the Jordanian government by the Iranian Advanced Persistent Threat (APT) group APT34 that used its own innovative version of this method.

The payload in the attack was a backdoor called Saitama, a finite state machine that used DNS to communicate. Our original article provides an educational deep dive into the operation of Saitama and is well worth a read.

Here we will expand on the tricks that Saitama used to keep its DNS tunelling hidden.

Saitama's DNS tunnelling

DNS is the Internet's "address book" that allows computers to lookup human-readable domain names, like `malwarebytes.com`, and find their IP addresses, like `54.192.137.126`.

DNS information isn't held in a single database. Instead it's distributed, and each domain has name servers that are responsible for answering questions about them. Threat actors can use DNS to communicate by having their malware make DNS lookups that are answered by name servers they control.

DNS is so important it's almost never blocked by corporate firewalls, and the enormous volume of DNS traffic on corporate networks provides plenty of cover for malicious communication.

Saitama's messages are shaped by two important concerns: DNS traffic is still largely unencrypted, so messages have to be obscured so their purpose isn't obvious; and DNS records are often cached heavily, so identical messages have to look different to reach the APT-controlled name servers.

Saitama's messages

In the attack on the Jordanian foreign ministry, Saitama's domain lookups used the following syntax:

`domain = message, counter '.'` root domain

The root domain is always one of `uber-asia.com`, `asiaworldremit.com` or `joexpediagroup.com`, which are used interchangeably.

The sub-domain portion of each lookup consists of a message followed by a counter. The counter is used to encode the message, and is sent to the command and control (C2) server with each lookup so the C2 can decode the message.

Four types of message can be sent:

1. Make contact

The first time it is executed, Saitama starts its counter by choosing a random number between `0` and `46655`. In this example our randomly-generated counter is `7805`.

The DNS lookup derived from that counter is:

`nbn4vxanrj.joexpediagroup.com`

The counter itself is encoded using a hard-coded base36 alphabet that is shared by the name server. In base36 each digit is represented by one of the 36 characters 0-9 and A-Z. In the standard base36, alphabet `7805` is written `60t` (`6` x `1296` + `0` x `36` + 30 x `1`). However, in Saitama's custom alphabet `7805` is `nrj`.

The counter is also used to generate a custom alphabet that will be used to encode the message using a simple substitution. The first message sent home is the command `0`, base36-encoded to `a`, which tells the server it has a new victim, prepended to the string `haruto`, making `aharuto`.

A simple substitution using the alphabet generated by the counter yields the message `nbn4vxa`.

a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 ↓ ↓ ↓ ↓ ↓ ↓ n j 1 6 9 k p b h d 0 7 y i a 2 g 4 u x v 3 e s w f 5 8 r o c q t l z m

The C2 name server decodes the counter using the shared, hard-coded alphabet, and then uses the counter to derive the alphabet used to encode `aharuto`.

It responds to the contact request with an IP address that contains an ID for Saitama to use in future communications. The first three octets can be anything, and Saitama ignores them. The final octet contains the ID. In our example we will use the ID `203`:

`75.99.87.203`

2. Ask for a command

Now that it has an ID from the C2 server, Saitama increments its counter to `7806` and signals its readiness to receive a command as follows: The counter is used to generate a new custom alaphabet, which encodes the ID, `203`, as `ao`. The counter itself is encoded using the malware's hard-coded base36 alphabet, to `nrc`, and one of Saitama's three root domains is chosen at random, resulting in:

`aonrc.uber-asia.com`

The C2 server responds to the request with the size of the payload Saitama should expect. Saitama will use this to determine how many requests it will need to make to retrieve the full payload.

The first octet of the IP address the C2 responds with is any number between 129 and 255, while the second, third and fourth octets signify the first, second, and third bytes of the size of the payload. In this case the payload will be four bytes.

`129.0.0.4`

3. Get a command

Now that it knows the size of the payload it will receive, Saitama makes one or more RECEIVE requests to the server to get its instructions. It increments its counter by one each time, starting at `7807`. Multiple requests may be necessary in this step because some command names require more than the four bytes of information an IP address can carry. In this case it has been told to retrieve four bytes of information so it will only need to make one request.

The message from Saitama consists of three parts: The digit 2, indicating the `RECEIVE` command; the ID `203`; and an offset indicating which part of the payload is required. These are individually base36-encoded and concatenated together. The resulting string is encoded using a custom base36 alphabet derived from the counter `7807`, giving us the message `k7myyy`.

The counter is encoded using the hard-coded alphabet to `nr6`, and one of Saitama's three root domains is chosen at random, giving us:

```
k7myyynr6.asiaworldremit.com
```

The C2 indicates which function it wants to run using two-digit integers. It can ask Saitama to run any of five different functions:

C2	Saitama
43	Static
70	Cmd
71	CompressedCmd
95	File
96	CompressedFile

Saitama functions

In this case the C2 wants to run the command `ver` using Saitama's `Cmd` function. (In the previous request the C2 indicated that it would be sending Saitama a four byte payload: One byte for `70`, and three bytes for `ver`.)

In its response, the C2 uses the first octet of the IP address to indicate the function it wants to run, `70`, and then the remaining three octets to spell out the command name `ver` using the ASCII codepoints for the lowercase characters "v", "e", and "r":

```
70.118.101.114
```

4. Run the command

Saitama runs the command it has been given and sends the resulting output to the C2 server in one or more DNS requests. The counter is incremented by one each time, starting at `7808` in our example. Multiple requests may be necessary in this step because some command names require more than the four bytes an IP address can carry.

```
p6yqqqqp0b67gcj5c2r3gn3l9epztnrb.asiaworldremit.com
```

The counter is encoded using the hard-coded alphabet to `nrb`, and one of Saitama's three root domains is chosen at random.

In this case the message consists of five parts: The digit 2, indicating the RECEIVE command; the ID `203`; and an offset indicating which part of the response is being sent; the size of the buffer; and a twelve-byte chunk of the output. These are individually base36-encoded and concatenated together. The resulting string is encoded using a custom base36 alphabet derived from the counter `7808`, giving us the message `p6yqqqqp0b67gcj5c2r3gn3l9epzt`.

Detection

Malwarebytes customers are protected from this attack via our Anti-Exploit layer. To learn more about the recent attack involving Saitama, read APT34 targets Jordan Government using new Saitama backdoor.

IOCs

Maldoc

Confirmation Receive Document.xls 26884f872f4fae13da21fa2a24c24e963ee1eb66da47e270246d6d9dc7204c2b

Saitama backdoor

update.exe e0872958b8d3824089e5e1cfab03d9d98d22b9bcb294463818d721380075a52d

C2s

uber-asia.com asiaworldremit.com joexpediagroup.com

SHARE THIS ARTICLE

ABOUT THE AUTHOR

Mark Stockley