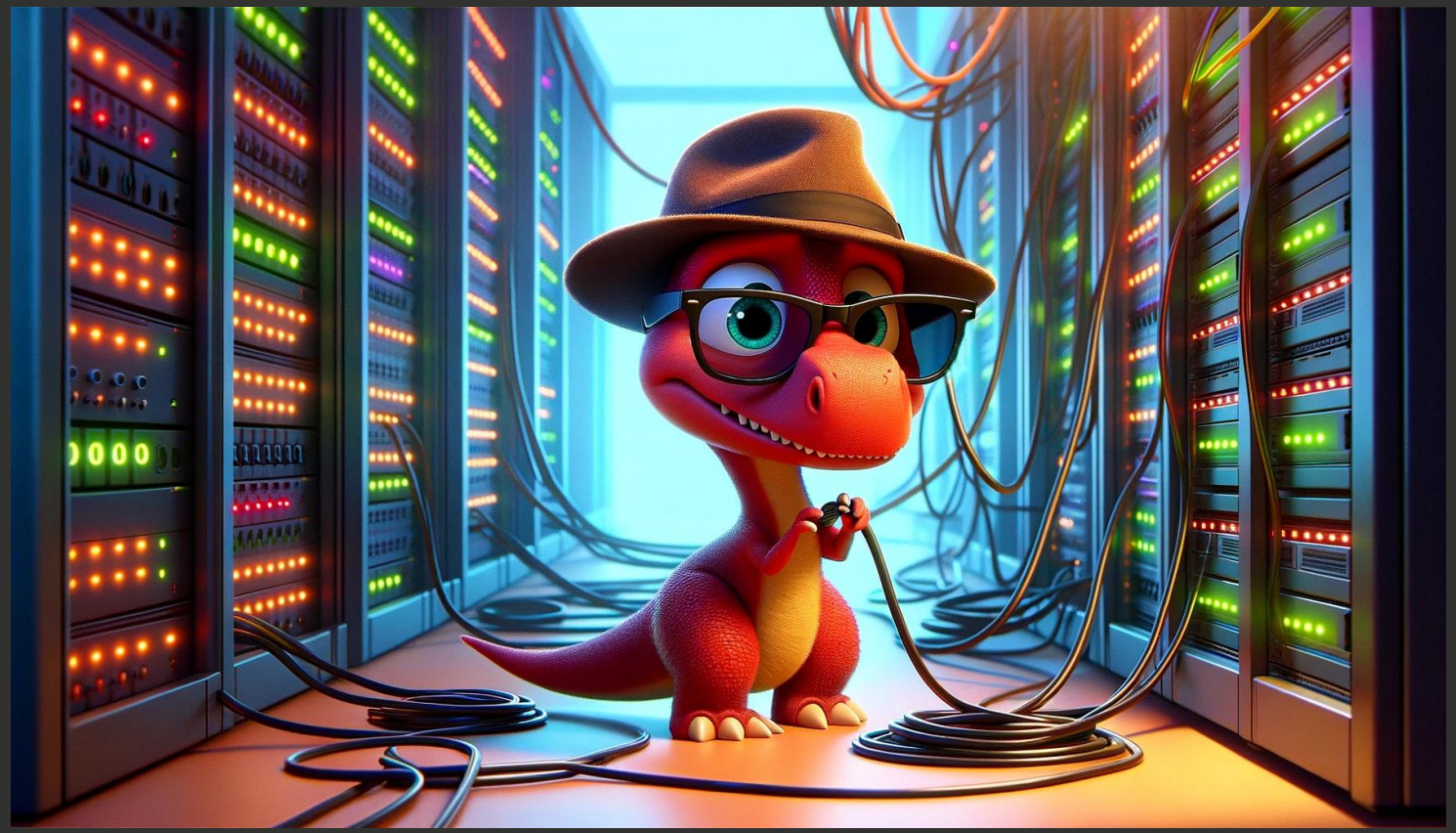
RED TEAM FAILS

“Oops, my bad I ruined the operation”
A story on how to fail a redteam

DrinkLoveShare – 2024

Build the infrastructure



- 1 Command and Control Server
  - DinoStrike
- 1 Redirector
  - Dinoginx
- 1 Phishing Server
  - HTTPS certificate
  - Domain: dinosrv.com

crt.sh Identity Search

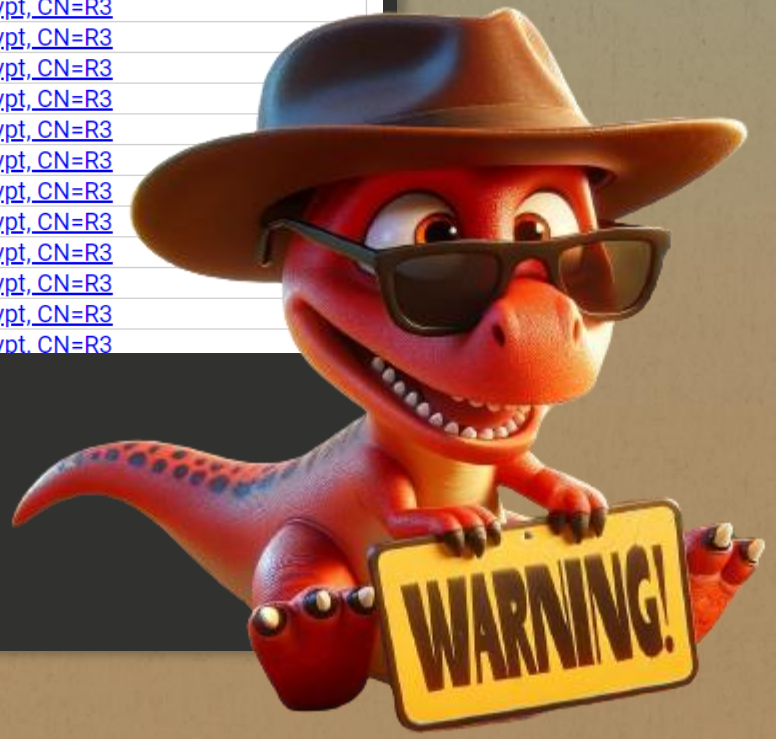
Group by Issuer

Criteria	Type: Identity Match: ILIKE Search: 'dinosrv.com'

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
11009475547	2023-11-06	2023-11-05	2024-02-03	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
11009484231	2023-11-06	2023-11-05	2024-02-03	meteorcorp.gophish.dinosrv.com	meteorcorp.gophish.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
10527468193	2023-09-07	2023-09-06	2023-12-05	dinostrike.dinosrv.com	dinostrike.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
10388712303	2023-09-07	2023-09-06	2023-12-05	microsoft.dinosrv.com	microsoft.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9890207977	2023-07-08	2023-07-08	2023-10-06	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9862123472	2023-07-08	2023-07-08	2023-10-06	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9356578500	2023-05-10	2023-05-09	2023-08-07	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9353676924	2023-05-10	2023-05-09	2023-08-07	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9356571822	2023-05-10	2023-05-09	2023-08-07	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
9354045539	2023-05-10	2023-05-09	2023-08-07	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
8896073401	2023-03-10	2023-03-10	2023-06-08	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3
8859830045	2023-03-10	2023-03-10	2023-06-08	www.dinosrv.com	www.dinosrv.com	C=US, O=Let's Encrypt, CN=R3

Oops, we leaked the infra in the certs...

- Use generic name for DNS
- Use wildcard for certificate (*)



Phishing attempt

```go
// RecipientParameter is the URL parameter that points to the result ID for a recipient.
const RecipientParameter = "rid"
```

```go
// ServerName is the server type that is returned in the transparency response.
const ServerName = "gophish"

```

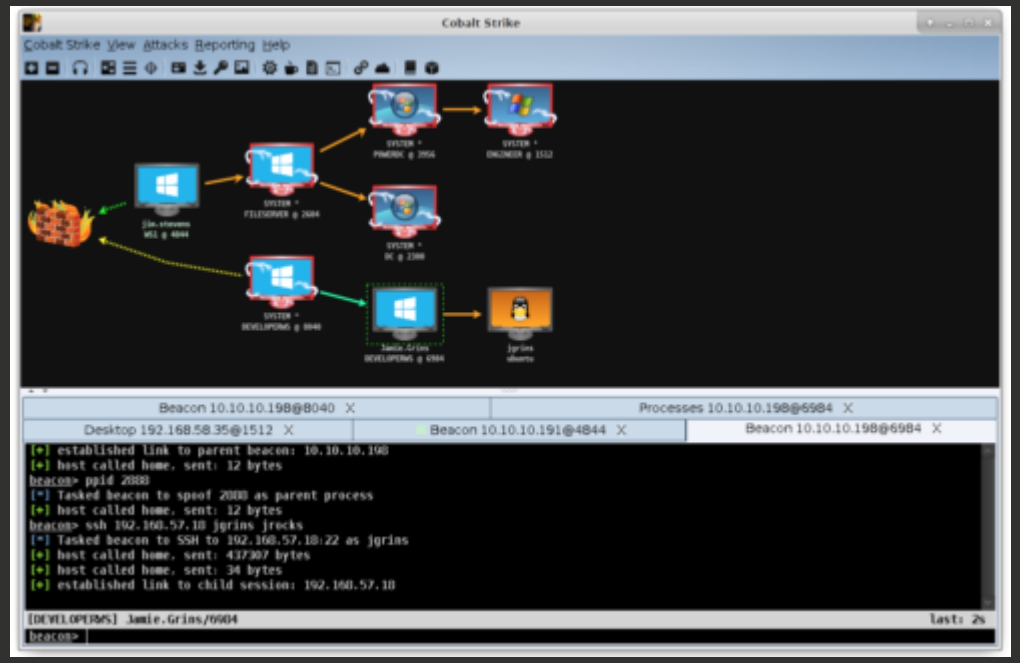
```go
// Add the transparency headers
msg.SetHeader("X-Mailer", config.ServerName)
if conf.ContactAddress != "" {
	msg.SetHeader("X-Gophish-Contact", conf.ContactAddress)
}

```

Oops, we used the default GoPhish binary with many IOC

- Customize your GoPhish
- Change the default parameter (rid)

Too much success ? very sus

Yeah, many callbacks



Cobalt Strike

Cobalt Strike View Payloads Attacks Site Management Reporting Help

external	internal	listener	user	computer	process	note	pid	arch	last	sleep
185.220....	10.1.40.12	DinoHTTP	Johnny Ca...	JOHNNYCAGE-PC	b859aeef79f485665ad...		1144	x86	4s	45 sec...
89.149.2...	10.127.0.177	DinoHTTP	Admin *	IMXSDNYJ	dino_http_x86.exe		1612	x86	3s	45 sec...
45.8.17.25	192.168.122.149	DinoHTTP	mike *	MIKE-PC	tmp_yvnioy6.exe		1864	x86	2m	45 sec...
195.164....	10.13.16.112	DinoHTTP	janusz *	JANUSZ-PC	malwar.exe		1912	x86	7s	45 sec...
185.220....	192.168.2.6	DinoHTTP	Joe Cage *	226533	nZyFCDljbD.exe		2004	x86	239ms	45 sec...
89.149.2...	10.127.0.170	DinoHTTP	Admin *	AILVMYUM	dino_http_x86.exe		2224	x86	2s	45 sec...
89.149.2...	10.127.0.163	DinoHTTP	Admin *	OZEMQECW	dino_http_x86.exe		2376	x86	7s	45 sec...
67.218.1...	192.168.243.144	DinoHTTP	0jDzBbE *	DBVD0teuSV	dino_http_x86.exe		2612	x86	878ms	45 sec...
89.149.2...	10.127.0.67	DinoHTTP	Admin *	EUCQOBEO	dino_http_x86.exe		4556	x86	17s	45 sec...

Event Log X Listeners X

```
12/21 22:13:03 *** initial beacon from Athena@192.168.1.70 (REVERSE)
12/21 22:17:30 *** initial beacon from admin@192.168.100.104 (USER-PC)
12/21 22:19:59 *** initial beacon from admin@192.168.100.70 (USER-PC)
12/21 22:26:30 *** initial beacon from Johnny Cage *@10.1.40.12 (JOHNNYCAGE-PC)
12/21 22:26:50 *** initial beacon from Admin *@10.127.0.177 (IMXSDNYJ)
12/21 22:26:51 *** initial beacon from mike *@192.168.122.149 (MIKE-PC)
12/21 22:27:15 *** initial beacon from Admin *@10.127.0.163 (OZEMQECW)
12/21 22:27:28 *** initial beacon from Admin *@10.127.0.170 (AILVMYUM)
12/21 22:27:35 *** initial beacon from janusz *@10.13.16.112 (JANUSZ-PC)
12/21 22:27:36 *** initial beacon from Joe Cage *@192.168.2.6 (226533)
12/21 22:27:58 *** initial beacon from jones *@192.168.2.4 (745773)
12/21 22:28:44 *** initial beacon from Admin *@10.127.0.67 (EUCQOBEO)
12/21 22:28:49 *** initial beacon from Joe Cage *@192.168.2.101 (936905)
12/21 22:28:59 *** initial beacon from 0jDzBbE *@192.168.243.144 (DBVD0teuSV)
[12/21 22:29] dino                                    [TeamServer IP:
event>
```

Who is Johnny Cage ?

```
################################################
## Staging process
################################################
## OPSEC WARNING!!!! Staging has serious OPSEC issues.
## It is recommed to disable staging and use stageless payloads
## Description:
##    Malleable C2's http-stager block customizes the HTTP staging process
##    Host payload for staging over HTTP, HTTPS, or DNS. Required by stagers.set
## Defaults:
##    uri_x86 Random String
##    uri_x64 Random String
##    HTTP Server Headers - Basic HTTP Headers
##    HTTP Client Headers - Basic HTTP Headers
set host_stage "true";
```

Oops, our payload is detected and we got SPAMMMMMED !

- Disable hosted payloads for staging purposes
- Never upload your binary on VirusTotal, or send the samples
- Geoblocking / IP whitelisting
- Guardrails using domain/computer/username

detection-rules / rules / windows / discovery_whoami_command_activity.toml

```
11    description = """
12    Identifies suspicious use of whoami.exe which displays user, group, and privileges information for the user who is
13    currently logged on to the local system.
14    """
```

```
process where event.type in ("start", "process_started") and process.name : "whoami.exe"
```

Oops, we executed the worst command

- Common detection trap, quick win for Blue Team: whoami

Graph all the things

```
<!-- This rule detects the creation of JSON files containing sensitive AD information
 obtained by the ingestor from the AD-->
<rule id="111155" timeframe="2" frequency="2" level="7">
  <if_sid>61613</if_sid>
  <field name="win.eventdata.image" type="pcre2">\.exe</field>
  <field name="win.eventdata.targetFilename" type="pcre2">(?i)([^\\]+?)(_computers\.json$|_domains\.json$|_ous\.json$|_users\.
  json$|_groups\.json$|_containers\.json$|_gpos\.json$)</field>
  <description>Possible Bloodhound activity detected: $(win.eventdata.targetFilename) file created by $(win.eventdata.image).</
  description>
  <mitre>
    <id>T1036</id>
  </mitre>
</rule>
```

rule.description

```
Zip file created: compressed data C:\\Users\\Attacker\\Desktop\\20230705064418_BloodHound.zip created by C:\\U
sers\\Attacker\\Desktop\\SharpHound.exe.
```

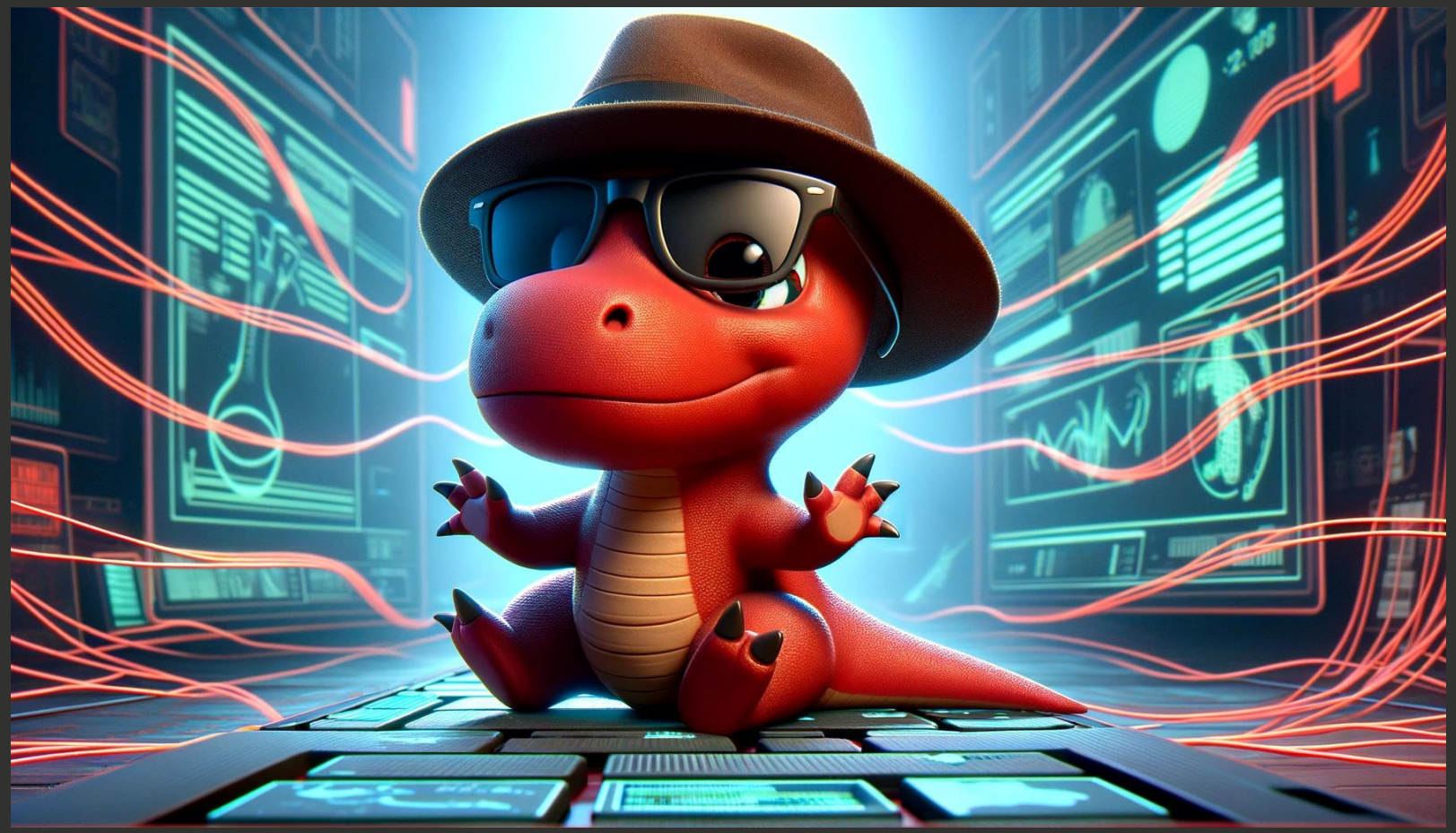
```
Possible Bloodhound activity detected: C:\\Users\\Attacker\\Desktop\\20230705064418_domains.json file created
by C:\\Users\\Attacker\\Desktop\\SharpHound.exe.
```

```
Possible Bloodhound activity detected: C:\\Users\\Attacker\\Desktop\\20230705064418_ous.json file created by
C:\\Users\\Attacker\\Desktop\\SharpHound.exe.
```

Oops, we used forgot the basics of opsec

- - Do not touch the disk (in memory execution only)
- - Encrypt your output data, at least reduce the leftovers

Kerberoasting



```python
    def kerberoasting(self):
        # Building the search filter
        searchFilter = "(&(servicePrincipalName=*)(UserAccountControl:1.2.840.113556.1.4.803:=512)"
        attributes = [
            "servicePrincipalName",
            "sAMAccountName",
            "pwdLastSet",
            "MemberOf",
            "userAccountControl",
            "lastLogon",
        ]
```

Oops, the query is too large

- CrackMapExec/NetExec LDAP queries use wildcard (*)

```
crackmapexec ldap $TARGETS -u $USER -p $PASSWORD --kerberoasting kerberoastables.txt
```

Move lat (like a ninja)

Oops,
no pivoting today

Impacket

- smbexec: BTOBTO or rand_char*8 svc
- psexec: RemComSvc
- wmiexec: cmd.exe /Q /c

sigma / rules / windows / builtin / system / service_control_manager / win_system_hack_smbexec.yml

Code | Blame | 33 lines (33 loc) · 1.11 KB | Raw

```yaml
title: smbexec.py Service Installation
id: 52a85084-6989-40c3-8f32-091e12e13f09
status: test
description: Detects the use of smbexec.py tool by detecting a specific service installation
references:
    - https://blog.ropnop.com/using-credentials-to-own-windows-boxes-part-2-psexec-and-services
    - https://github.com/fortra/impacket/blob/33058eb2fde6976ea62e04bc7d6b629d64d44712/examples
    - https://github.com/fortra/impacket/blob/edef71f17bc1240f9f8c957bbda98662951ac3ec/examples
author: Omer Faruk Celik
date: 2018/03/20
modified: 2023/11/09
tags:
    - attack.lateral_movement
    - attack.execution
    - attack.t1021.002
    - attack.t1569.002
logsource:
    product: windows
    service: system
detection:
    selection_eid:
        Provider_Name: 'Service Control Manager'
        EventID: 7045
    selection_service_name:
        ServiceName: 'BTOBTO'
    selection_service_image:
        ImagePath|contains:
            - '.bat & del '
            - '__output 2^>^&1 >'
    condition: selection_eid and 1 of selection_service_*
falsepositives:
    - Unknown
level: high
```

impacket / examples / smbexec.py

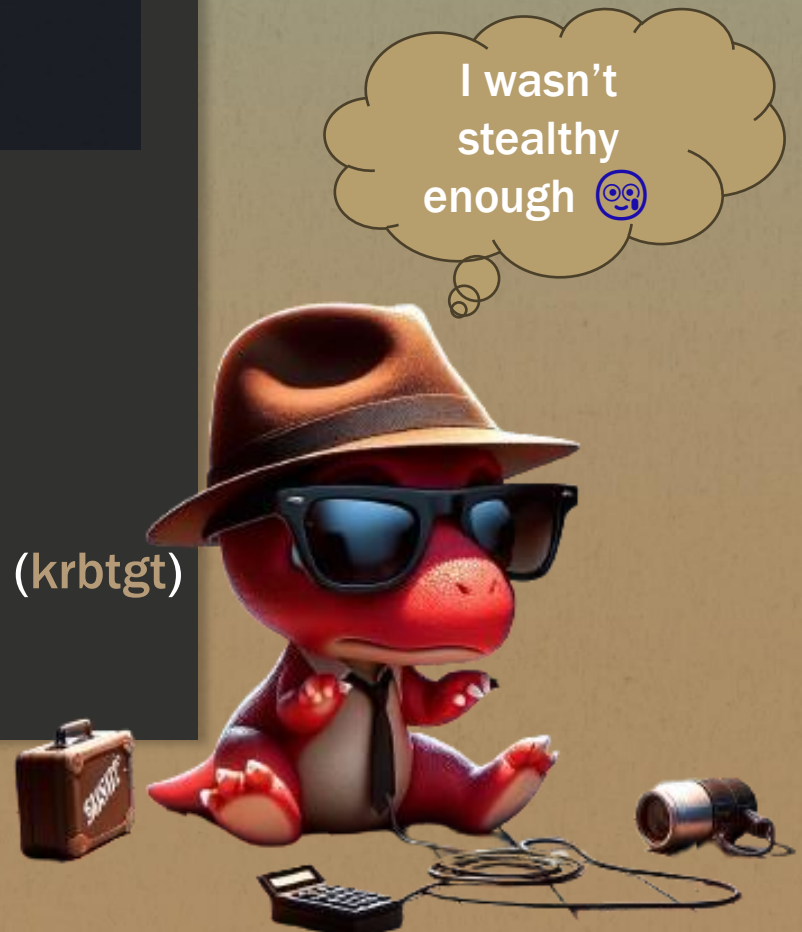
```
DUMMY_SHARE     = 'TMP'
SERVICE_NAME    = 'BTOBTO'
CODEC = sys.stdout.encoding
```

```
' -p '[redacted]' -d [redacted] --ntds
[*] Windows 10.0 Build 20348 x64 (name:DC) (domain:[redacted]) (signing:True) (SMBv1:False)
[+] [redacted]
[+] Dumping the NTDS, this could take a while so go grab a redbull...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:[redacted]:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:[redacted]:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:[redacted]::
```

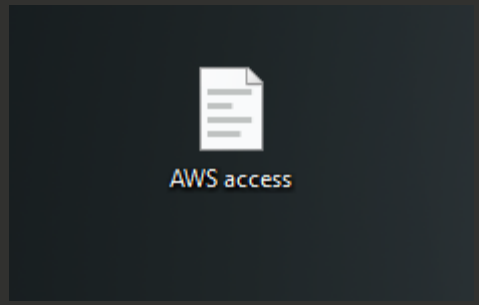
Oops, I looted too many hashes !

Looting NTDS via CME/NXC

- Replication is always done by Domain Controller (Computer account)
- Do you really need to dump ALL the users and computers ? One is enough (krbtgt)



The best view is from the clouds ☁️




```
└$ aws configure
AWS Access Key ID [None]: ILoveSecrets
AWS Secret Access Key [None]: YouLikeThem?
Default region name [None]: DinoWorld
Default output format [None]:
```

Oops, GuardDuty was watching

```
└─$ grep -ri platform.release .
./useragent.py:            platform_version=platform.release(),
./session.py:            f'{platform.system()}/{platform.release()}'
```

```
Python 3.11.6 (main, Oct
Type "help", "copyright",
>>> import platform
>>> platform.release()
'6.5.0-kali3-amd64'
```

PenTest:IAMUser/KaliLinux

An API was invoked from a Kali Linux EC2 machine.

Default severity: Medium

- **Data source:** CloudTrail management event

This finding informs you that a machine running Kali Linux is making API calls using credentials that belong t AWS account in yo
Kali Linux is a popular penetration testing tool that security professionals use to identify weaknesses in quire patching. A
this tool to find EC2 configuration weaknesses and gain unauthorized access to your AWS environmen

GAME OVER - See you in jail !

Dino: 0
Blue Team: 1

No question please, I have bad advices

@pentest_swissky