Okechukwu Ude
Supervisor: Dr. Bobby Swar

Securing Remote Access Networks Using Malware Detection Tools for Industrial Control Systems

Concordia University of Edmonton
Faculty of Management

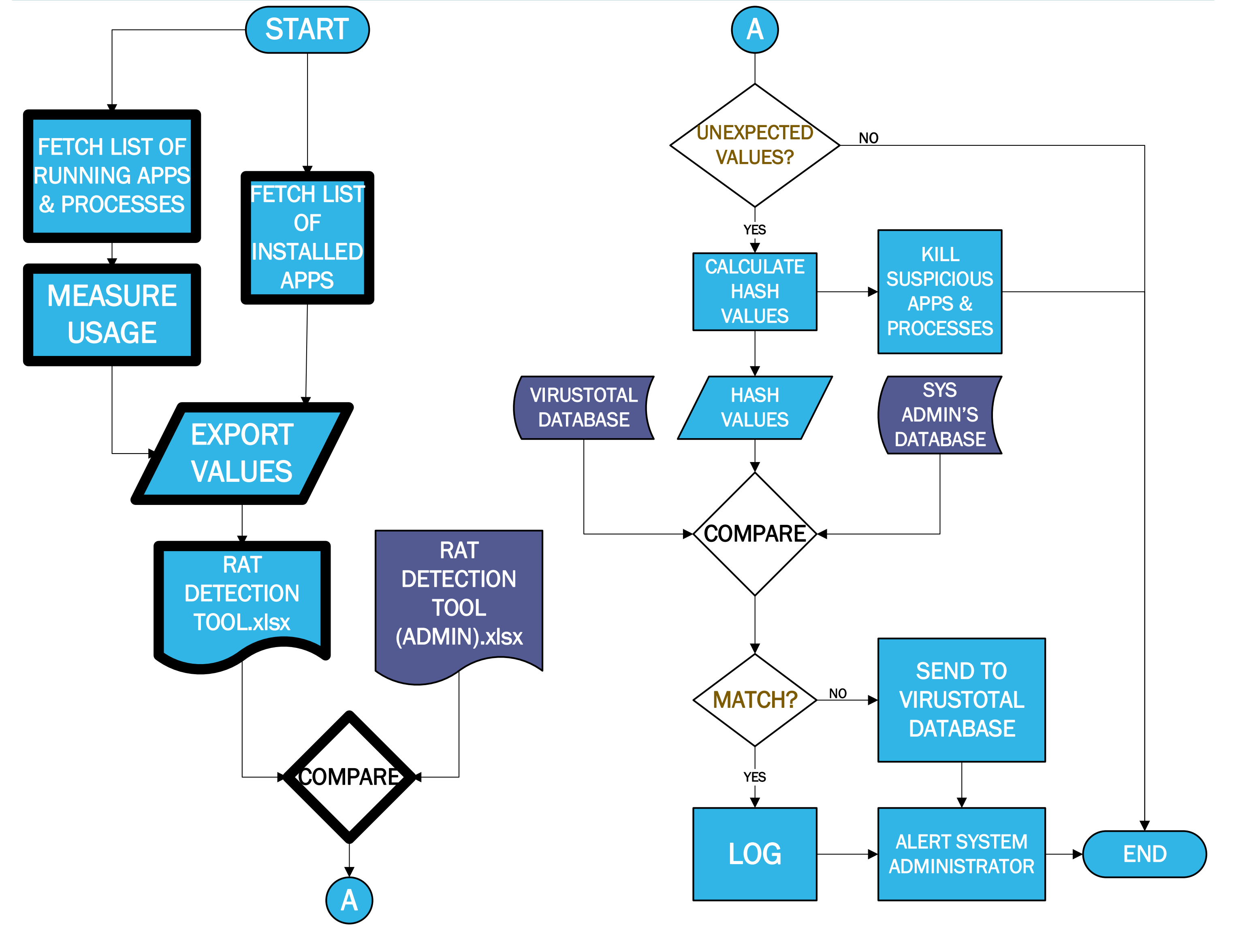
INTRODUCTION

- Remote Access Trojans (RATs) are a form of malware which grant an attacker administrative access to a remote device, allowing covert surveillance, together with unfettered access, thereby establishing a foothold in the target system.
- Most cyberattacks on Industrial Control Systems (ICS) are launched using RATs.
- According to a Department of Homeland Security report, at least 55% of the 245 reported ICS attack cases in 2015 were attributed to RATs.
- These statistics underline the need to increase the detection strength of malware-detection implementations.
- The tool was encoded in Python in line with cybersecurity best practices.

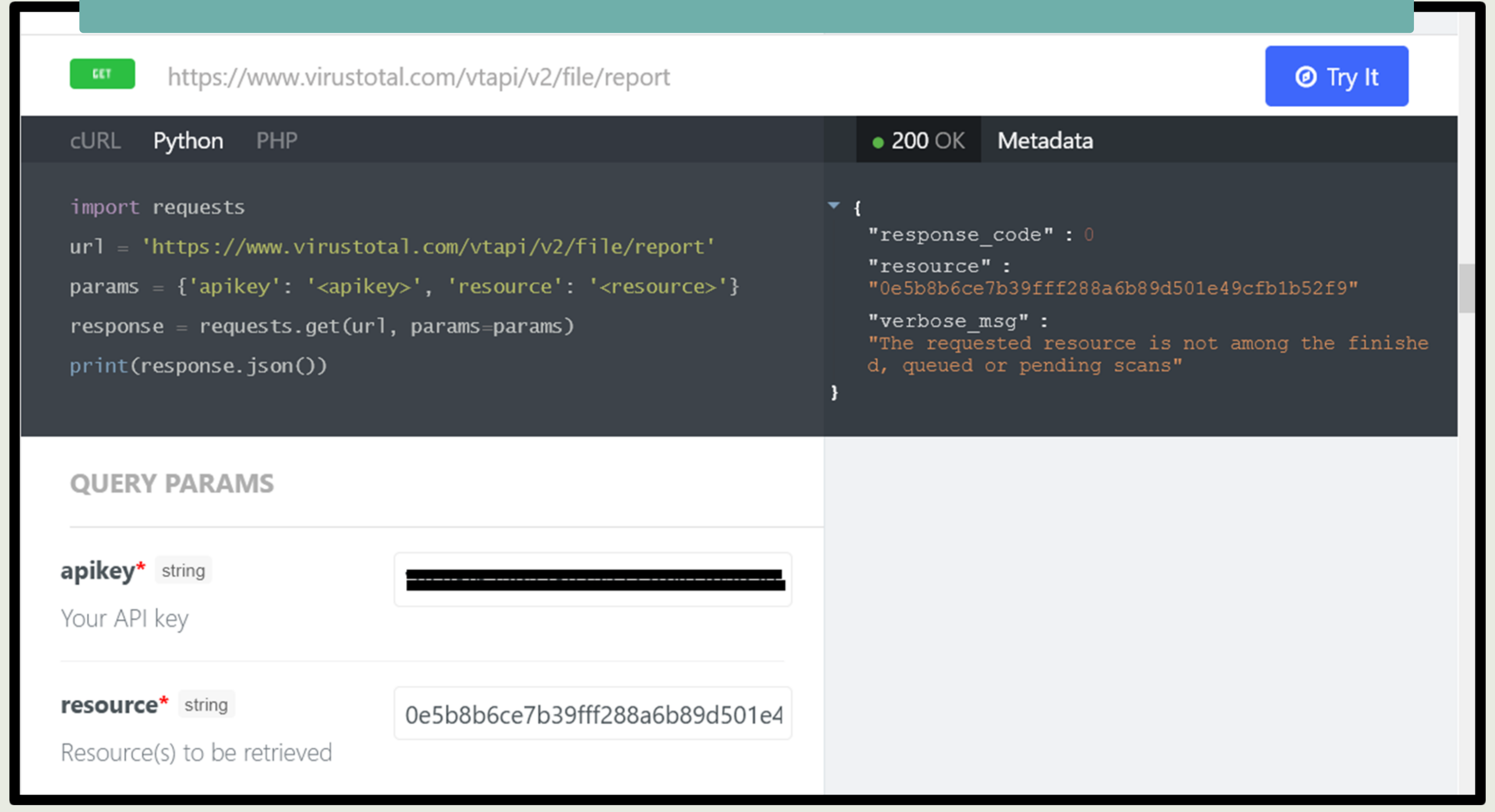
OBJECTIVES

- Development of a tool for the detection and elimination of RATs in ICS.
- This tool - unlike antivirus software which compares applications against databases of known threats, analyzes process patterns of running applications and processes, then flags suspicious applications or processes for further analysis.

METHODS


START
FETCH LIST OF RUNNING APPS & PROCESSES
MEASURE USAGE
FETCH LIST OF INSTALLED APPS
EXPORT VALUES
RAT DETECTION TOOL.xlsx
RAT DETECTION TOOL (ADMIN).xlsx
COMPARE
A
A
UNEXPECTED VALUES?
NO
YES
CALCULATE HASH VALUES
KILL SUSPICIOUS APPS & PROCESSES
VIRUSTOTAL DATABASE
HASH VALUES
SYS ADMIN'S DATABASE
COMPARE
MATCH?
NO
SEND TO VIRUSTOTAL DATABASE
YES
LOG
ALERT SYSTEM ADMINISTRATOR
END

RESULTS



```
import requests
url = 'https://www.virustotal.com/vtapi/v2/file/report'
params = {'apikey': '<apikey>', 'resource': '<resource>'}
response = requests.get(url, params=params)
print(response.json())
```

```
{
  "response_code" : 0
  "resource" :
  "0e5b8b6ce7b39fff288a6b89d501e49cfb1b52f9"
  "verbose_msg" :
  "The requested resource is not among the finished, queued or pending scans"
}
```

QUERY PARAMS

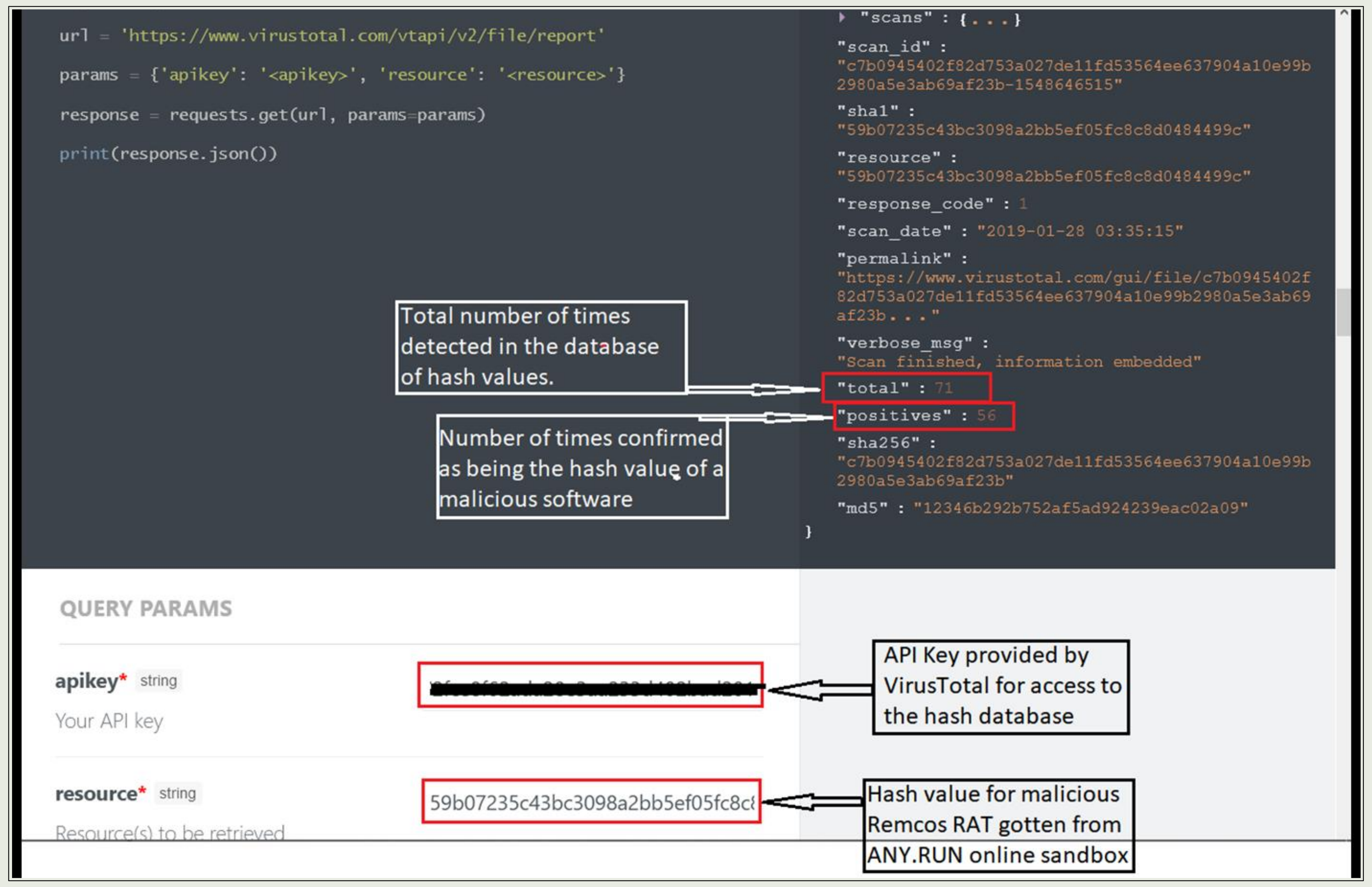
apikey* string
Your API key
[redacted]

resource* string
Resource(s) to be retrieved
0e5b8b6ce7b39fff288a6b89d501e4

Figure 1: VirusTotal Hash Scan Result for the Legitimate Stub

S/ No	Family Name	Comparison Scan Status	Hash Value (SHA1)	Hash Check Result	
				Sys Admin	*Virus Total*
1	Remcos RAT (Legitimate)	Detected (remcos_agent.exe)	0e5b8b6ce7b39fff288a6b89d501e49cfb1b52f9	1	0
2	Remcos RAT (malicious)	Detected (remcos.exe)	59b07235c43bc3098a2bb5ef05fc8c8d0484499c	0	1

Table 1: Results of the Hash Check on Tested Samples



```
url = 'https://www.virustotal.com/vtapi/v2/file/report'
params = {'apikey': '<apikey>', 'resource': '<resource>'}
response = requests.get(url, params=params)
print(response.json())
```

```
  "scans" : {...}
  "scan_id" :
  "c7b0945402f82d753a027de11fd53564ee637904a10e99b2980a5e3ab69af23b-1548646515"
  "sha1" :
  "59b07235c43bc3098a2bb5ef05fc8c8d0484499c"
  "resource" :
  "59b07235c43bc3098a2bb5ef05fc8c8d0484499c"
  "response_code" : 1
  "scan_date" : "2019-01-28 03:35:15"
  "permalink" :
  "https://www.virustotal.com/gui/file/c7b0945402f82d753a027de11fd53564ee637904a10e99b2980a5e3ab69af23b..."
  "verbose_msg" :
  "Scan finished, information embedded"
  "total" : 71
  "positives" : 56
  "sha256" :
  "c7b0945402f82d753a027de11fd53564ee637904a10e99b2980a5e3ab69af23b"
  "md5" : "12346b292b752af5ad924239eac02a09"
}
```

Total number of times detected in the database of hash values.

Number of times confirmed as being the hash value of a malicious software

QUERY PARAMS

apikey* string
Your API key
[redacted]

API Key provided by VirusTotal for access to the hash database

resource* string
Resource(s) to be retrieved
59b07235c43bc3098a2bb5ef05fc8c

Hash value for malicious Remcos RAT gotten from ANY.RUN online sandbox

Figure 2: VirusTotal Hash Scan Result for the Malicious Stub

S/ No	Family name	Programming Language	Debut Year	Comparison Scan Result
29	NjRAT	.NET	2012	server.exe
30	Njworm	Visual Basic	2013	njworm.exe
31	NovaLite	Delphi	2011	Server.exe
32	Nuclear	Delphi	2003	Server.exe
33	Orion	Delphi	2014	orionserver.exe

Table 2: Results of the Comparison Scan on Tested Samples

DISCUSSION/CONCLUSION

- The research began with a comparative analysis of some open-source RAT detection tools.
- A high-level representation of the deficiencies of the identified detection tools was then provided using a gap analysis approach. The summation of the identified gaps points out the absence of a host-based, process hash-checking functionality, hence the need for a hash-checking section in the source code.
- This research identified and addressed this unavailability and developed a solution that can detect malicious processes in a system.
- The single-script format of the source code leaves little room for attackers to exploit interconnection points in the mechanism.

RELEVANCE

- This research is expected to contribute knowledge towards increasing the efficiency of existing remote access trojan detection methods.
- The use of the created RAT detection tool will enhance the efficiency and effectiveness of ICS security measures implemented across all critical infrastructure sectors.

SPONSORS

Concordia Research Committee

Faculty of Management, Concordia University of Edmonton

ADVISORS

Dr. Bobby Swar

ACKNOWLEDGEMENTS

I would like to extend my appreciation to:

Dr. Bobby Swar

Engr. Ikenna Ude

Ana Tinoco, and all the good people at VirusTotal and ANYRUN