Simulating Malware Communications in Distributed Networks

Fatih Ozavci (https://linkedin.com/in/fozavci)

Executive Summary

Identifying malware communications in distributed networks is always challenging due to data volume, velocity and variety. Nowadays, malware communications use various network protocols with resiliency improvements such as traffic profiling, domain fronting and leveraging legitimate cloud services. On the other hand, traditional security controls work individually with a single scope such as endpoint activities, network ingress/egress filtering or proxying communications. Due to this mismatch, they would be insufficient to detect modern Command and Control (C2) communications.

Cyber Security Analytics (CSA) is used for collecting related network and endpoint security control logs to achieve meaningful detections with deep learning. They offer rule-based, signature-based and deep learning based detections for the data pool managed. This research provides a list of malware communications, their use by threat actors, and indicators of compromise which can be used to build detections. It also provides a threat modelling to prioritise certain communications where the budget or implementation is limited. CSA also needs verification and reliability checks which are generally provided with malware traffic replays or known security tools. As this approach has numerous limitations; adversary simulations are being used to generate malware activities to verify efficiency of CSA detections.

Adversary simulations in distributed networks also come with their own challenges to generate malware communications. For example, Petaq C2 developed by the author, or Caldera/Sandcat developed by Mitre, can be used to generate malware activities and communications as a cyber-attack scenario. Though, they support only a number of protocols with no profiling. The open source and commercial tools used by threat actors also have their own individual protocol and profile limitations (e.g. Cobalt Strike and External C2). In addition, the offensive tools are not easily accessible by the defenders due to their hostile features (e.g. compliance, regulation). TA505+ adversary simulation pack developed by the author demonstrates the difficulties of deploying a full-scale adversary simulation, and its coding requirements.

Tehsat malware generator is developed as a response to simplify identifying the CSA gaps listed above with a focus on malware communications instead of all malware activities. Tehsat is capable to generating malware traffic on HTTP(S), TCP, UDP and Websocket protocols with profiling. This paper provides a threat intelligence driven malware communications scenario simulated by Tehsat to demonstrate some of its features. The phases of the malware traffic generation and the tool accompany to the research. In a multi cloud deployment of Tehsat as suggested, a scenario-based malware traffic can be simulated in distributed networks. Using this approach, it’s possible to identify the gaps of CSA implemented in larger networks, analyse its efficiency against cutting-edge distributed C2 techniques, and improve CSA detections.

Table of Contents

Malware Communication Channels and Traditional Detections

Command & Control (C2) services are designed to manage malicious software and compromised systems remotely. While HTTP(S) protocol is widely used for the C2 activities, other protocols such as TCP, ICMP, DNS and SMB are also used in local networks or malware registration activities over Internet. In addition, threat actors also use social media, images and previously compromised services as C2 channels. In my previous research, **Current State of Malware Command and Control Communication Channels and Future Predictions**[i], I described widely used C2 techniques with some future estimations in details. Some interesting C2 channels used by threat actors are below.

- Turla APT group using Instagram comments to dynamically resolve the C2 servers [ii],
- Platinum APT group using steganography[iii],
- OceanLotus APT group using steganography for payload delivery [iv],
- APT34 using DNS over HTTPS as C2 channel[v].

C2 protocols running on various channels also have different purposes which may change the data size, delivery type or protocol specifications. Limited data size for malware registration or C2 discovery, slow interval checks for long term access, faster communications for real-time access, or autonomous controls on malware are examples of common C2 activities. The "**Malware Communications and Indicators of Compromise**" appendix shows main categories of these activities and their Indicators of Compromise (IoC) which can be used to build detections.

Detecting malicious behaviours in a reverse engineering or malware analysis exercise would be relatively easy using the given IoCs. However, it's way harder to detect same activities in a large network generating big data and telemetry. Traditional security controls work individually with a single scope such as endpoint activities, network ingress/egress filtering or proxying communications instead of correlation of events. Encryption is also a challenge as it's widely used for HTTP(S), TCP/TLS and HTTP API responses, but cannot be decrypted due to compliance and regulations. Finally, placing the security controls as border controllers doesn't provide sufficient visibility for distributed networks.

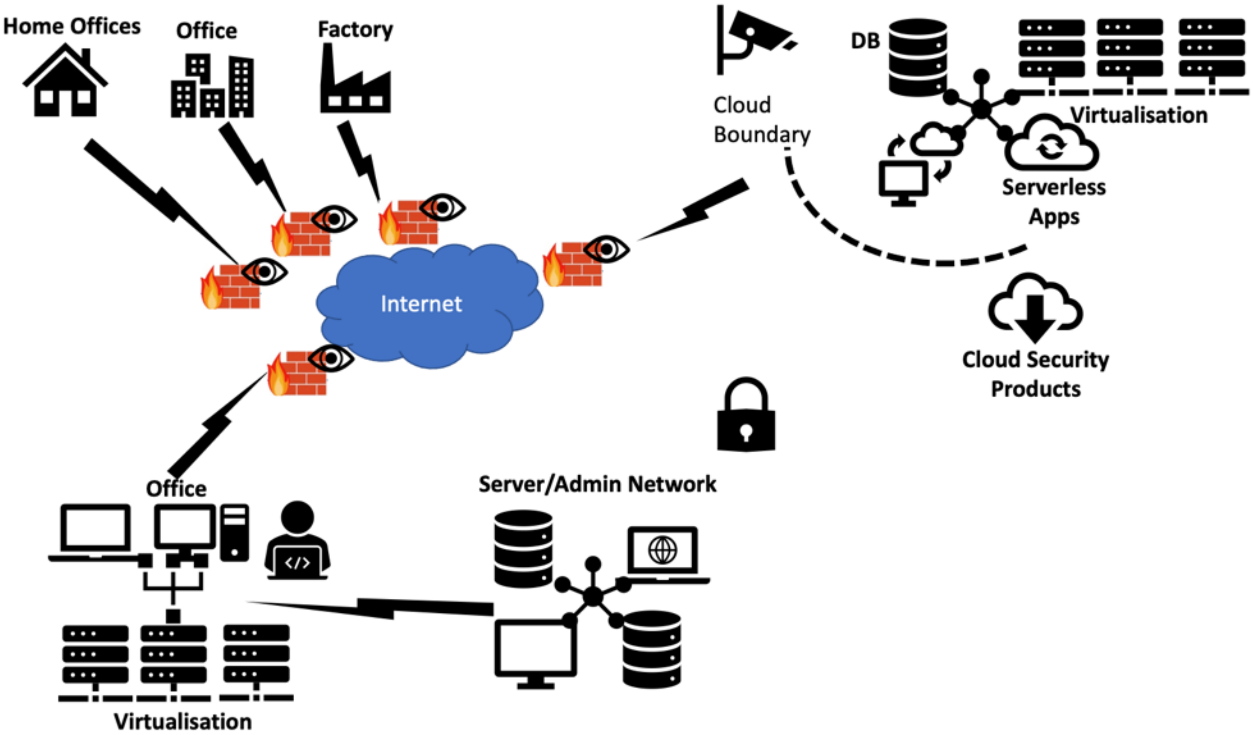


Figure 1 Complexity of Distributed Networks

The Figure 1 shows how security controls would be placed in a complex network in general. As seen in the diagram, the firewalls and proxies would stop the malicious Internet activities outgoing/incoming, but they have no use in internal networks. If the traffic or data transferred over internet is encrypted, they would only pass them through. The cloud vendors provide better visibility for the virtualised environments and serverless applications. However, this also comes with a caveat as they're not enabled by default, and also expensive.

Hence, the threat actors design their C2 communications based on the traditional monitoring weaknesses such as lack of internal network monitoring, lack of TLS interception, limited HTTP(S) anomaly detections and missing security controls due to their cost. As David Fifield et al. [vi] introduced, domain fronting is also a great way to circumvent traditional security controls and censorship. If a threat actor combines domain fronting, legitimate cloud services and proper encryption; it's highly difficult to detect C2 communications with traditional security controls.

Malware Traffic in Cyber Data Analytics Era

In a complex network such as the Figure 1, malware activities would be running on corporate internet access, virtualised networks, cloud internal networks, cloud internet access, and various segments of the corporate network. Figure 2 shows sample malware traffic types per networks; such as HTTP, social media and DNS services are generally used for malware to C2 server communications, but malware to malware communications are generally based on raw TCP, SMB Named Pipes or SSH-like services.

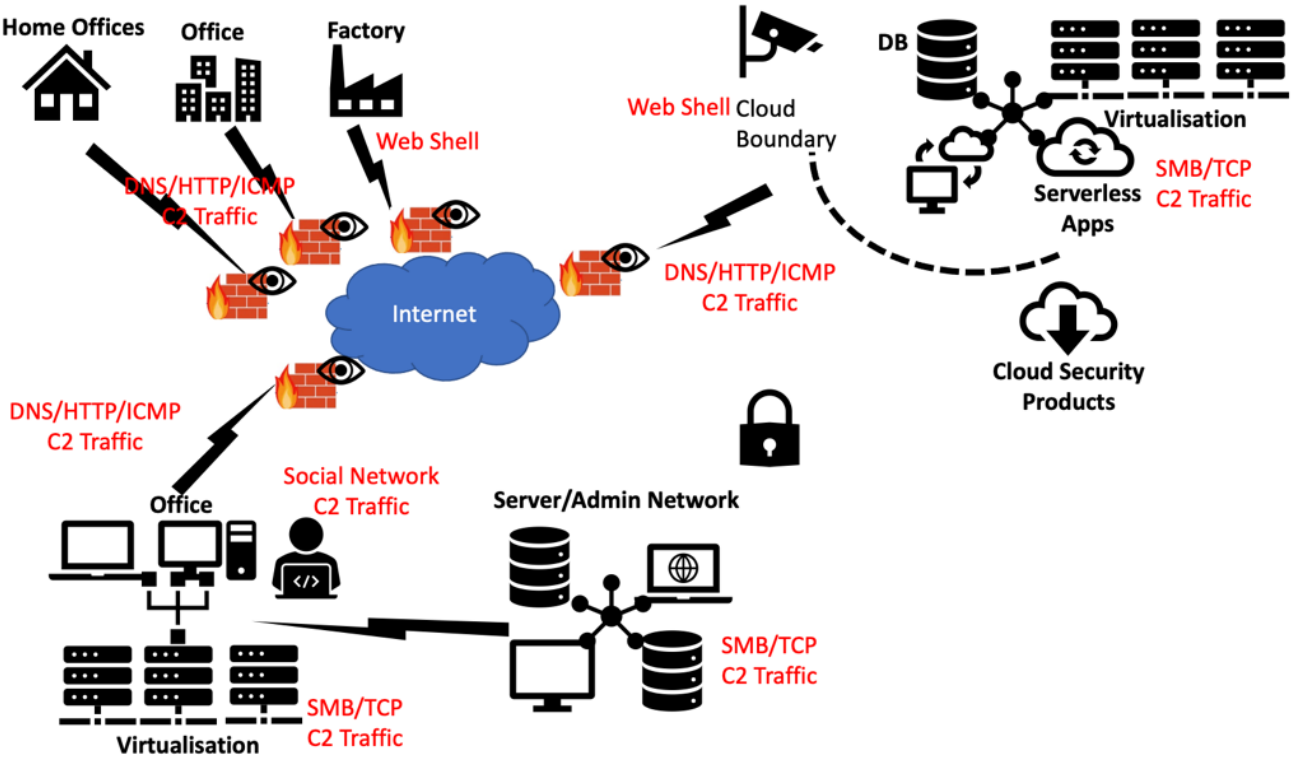


Figure 2 Malware Network Communications

Designing the mitigations for all networks would be an unreachable goal due to cost management and technical limitations. Threat modelling would assist us to prioritise potential malicious communications, and deploy the solutions or mitigations accordingly. Therefore, this paper includes **A Sample Threat Modelling for Malware Communications** appendix. The sample threat modelling suggests certain type of malware communications to be prioritised over others, and specific mitigations per communication type.

Data collection about malware traffic and activities require security engineers to work with data scientists due to telemetry requirements. Data Analytics (DA) and Machine Learning (ML) are used to process the big data generated by security controls such as network firewalls, network-based intrusion detection systems, TLS terminators, endpoint detection and response products, and cloud security products.

Cyber Security Analytics (CSA) is designed to collect relevant events from the key data sources as above, and process them using rule-based detections, ML based detections, and vendor supported signature-based detections. Rule-based detections are used for instant response to known activities such as detecting a prioritised known-bad traffic identified by the Cyber Threat Intelligence (CTI) team. Signature-based detections are designed to identify known-bad traffic in general using a feed supplied by CTI vendors or Cyber Response (CR) vendors. Unlike rule or signature-based detections, ML-based detections are designed to identify anomalies while generalising the activities and spotting the non-normal activities.

Academic researchers also suggest a variety of deep learning techniques to identify malicious activities in large networks. For example, according to Dutta V. et al[vii], certain deep learning models (e.g. Deep Neural Network [DNN], Long Short-Term Memory [LSTM]) are highly efficient to identify malicious activities on network traffic generated by critical infrastructure systems and internet of things (IoT). Stergiopoulos G. et al[viii] suggest to utilise ML algorithms (more specifically CART and KNN algorithms) to differentiate malicious activities in TCP/IP communications. In their research, they claim that they successfully identified malicious activities in unencrypted traffic, and even encrypted traffic which is highly critical in modern CSA environments.

Designing data sources (e.g. endpoint security, cloud networks, virtualised networks, proxies), data ingestion services and integrating them into the corporate networks are as important as how CSA detections designed. The key requirements are collecting the correct data related to ingress/egress filtering, efficient interception of encrypted communications, collecting cloud data for network activities, internal network monitoring, endpoint to endpoint communication monitoring, and finally interpreting the HTTP communications processed by proxies. The data coming from the sources should be as fast as possible due to the response time and correlation requirements of the CSA environments. The performance of the data ingestion could be improved with optimising the pre-processing and normalisation of data as well.

The CSA environments should be also tested periodically to identify weak spots, detections or performance issues. The Mitre Att&ck framework[ix] can be used build these test cases to verify the CSA efficiency. Threat actors use open source or commercial C2 servers to perform their attacks, such as Cobalt Strike[x], Merlin C2[xi], Godoh C2[xii], Metasploit Framework[xiii] and PowerShell Empire[xiv]. Hence, the very same tools can be used to generate the malicious activities. On the other hand, preparing a cyber-attack scenario to simulate a set of malicious activities and integrating them to the defence solutions is challenging.

Scenario Based Malware Traffic Generation

Adversary simulations are widely used to improve data analytics as they're more realistic than generating random malware traffic. In their research, Strom at al. [xv] summarise data analytics use for detecting various C2 activities such as payload delivery, interactive communications and scenario development using Mitre Att&ck framework. Though, the weakness of their research is they have no suggestions to automate the simulations used.

Applebaum et al. [xvi] [xvii] [xviii] [xix] released a set of adversary simulation research papers with a focus of automating simulations to improve data analytics and deep learning for better incident detections. One of their major contributions is scenario-based automation tools called Mitre Caldera[xx] and Sandcat. However, their work and tools also have a gap which is only limited number of communication types to be simulated. Caldera and Sandcat support only a limited number of protocols with limited customisations. Therefore, defenders have limited options to build a large scenario to simulate various C2 communications used for different stages of the attacks. Another challenge was also to be able to run real-life malware or tools in simulations as Caldera doesn't provide any support for it.

As I'm inspired of their work, and also try to mitigate some of these gaps, I developed Petaq C2[xxi] to provide scenario-based adversary simulation exercises to efficiently test CSA detections in large networks. To demonstrate Petaq's malware and C2 capabilities, I also prepared a complete TA505+ Adversary Simulation Pack[xxii] coming with cutting edge security bypasses and custom malware codes. This Threat Intelligence driven pack includes the TA505 tradecraft upgraded against cutting-edge defence systems to understand their impact in highly protected environments. I also presented it in the Purple Team Summit[xxiii] with practical scenario automations.

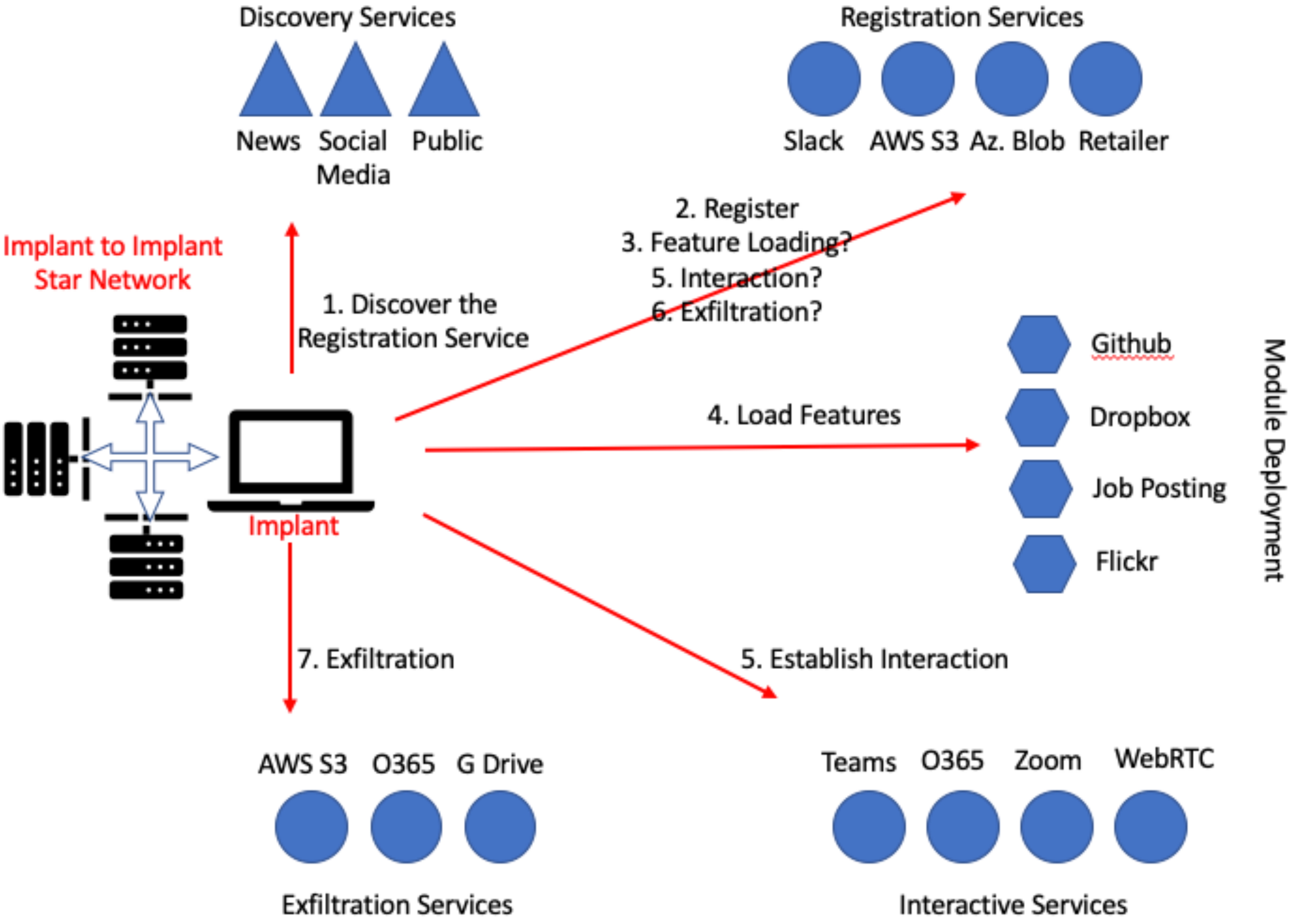


Figure 3 Distributed Command & Control Services

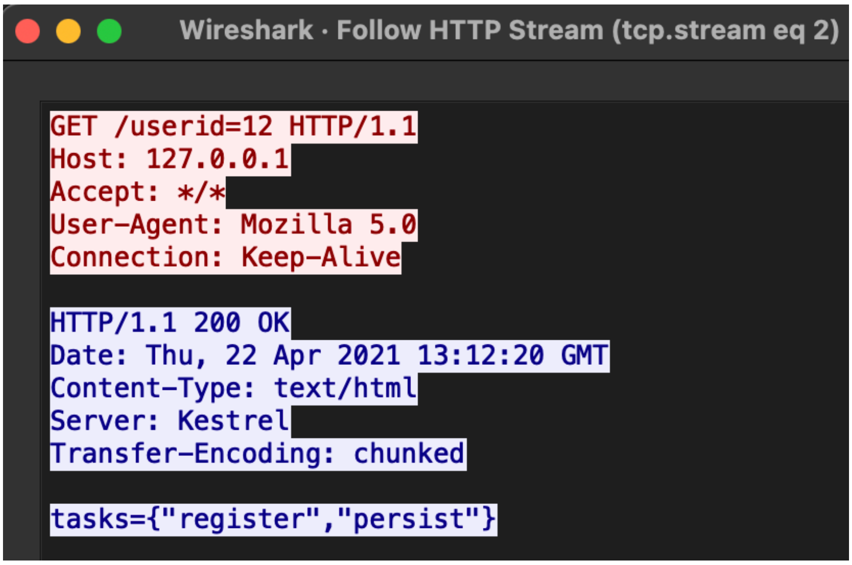
In near future, we may start encountering distributed C2 infrastructure used by threat actors such as Figure 3 taken from my previous paper (i). Carvalho M. et al.[xxiv] also suggests a human agent teamwork supported C2 for moving target defence (MTD) approach, though, it's only applicable to limited areas. CSA would provide efficient detections to identify most of these hypothetical communications. On the other hand, there will be challenges such as sensor placements, handling encryption, data ingestion speed and effectiveness of the algorithms. The best way to discover these gaps in CSA is to simulate the traffic from various locations of the networks. Unfortunately, Petaq C2 and Mitre Caldera won't be sufficient to provide this level of variety due to their C2 communication capabilities. Hence, I recently developed **Tehsat - Malware Traffic Generator** which can be used for simulating various C2 protocols without hostile activities. The alpha release prepared for this paper has only HTTP, Websocket, TCP and UDP protocols supported due to the time constraints.

Tehsat has profile-based service management approach for C2 channel and implant generation. The simulation starts with creating a profile, then creating a service based on it, and finally creating an implant for the service to simulate that C2 communication. I used IceID malware and its Cobalt Strike communication as an example for this basic simulation. E. Hjelmvik's blog post about IceID and Cobalt Strike malware traffic analysis[xxv] shows sufficient information for us to generate a profile. Even though Tehsat has more options, to keep the sample traffic as simple as possible, I extracted the URLs and HTTP methods in Figure 4 from the blog.

HTTP Method	URI
GET	http://lesti[.]net/userid=
POST	http://lesti[.]net/update.php
GET	https://45.147.229[.]157/match/
POST	https://45.147.229[.]157/submit.php

Figure 4 IceID - Cobalt Strike URIs

In **Tehsat - Malware Traffic Generator** appendix, I explained the actions taken while creating the profile, running services, generating implant, simulating traffic and monitoring the results step by step. When the implant compiled and executed, it will connect to the C2 services designed and configured. The implant compiled can be mass-deployed to the designated victims using PowerShell or group policies as it's a .NET 5 assembly. Figure 5 shows the sample traffic generated for the first IcedID URI; the read is sent by implant representing the malware, and blue is the tasks coming from the C2 server. So, the steps explained in the appendix can be repeated for each URI or service to generate different channels at the same time.



```
GET /userid=12 HTTP/1.1
Host: 127.0.0.1
Accept: */*
User-Agent: Mozilla 5.0
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Thu, 22 Apr 2021 13:12:20 GMT
Content-Type: text/html
Server: Kestrel
Transfer-Encoding: chunked

tasks={"register","persist"}
```

Figure 5 Tehsat – Sample Malware Traffic Generated

Tehsat's future releases will support implant management protocols, binary and text communication types, additional protocols (e.g. SMB, DNS, DNS over HTTPS, ICMP) and better reporting features. Through a cloud integrated deployment of Tehsat, it would be possible to generate various malware activities to train or test the data analytics platforms. Figure 6 shows a distributed deployment example with two cloud vendors and 5 redirectors used to simulate various malware traffic at the same time.

This example scenario would be;

1. APT1 threat actor deploys malware, manages with DNS, and resells the victims to APT39,
2. APT39 uses a HTTP profile for long term beaconing, and HTTP(S) for interactive communications,
3. ICMP and HTTP Serverless protocols are used for proxy and tunnelling by APT39.

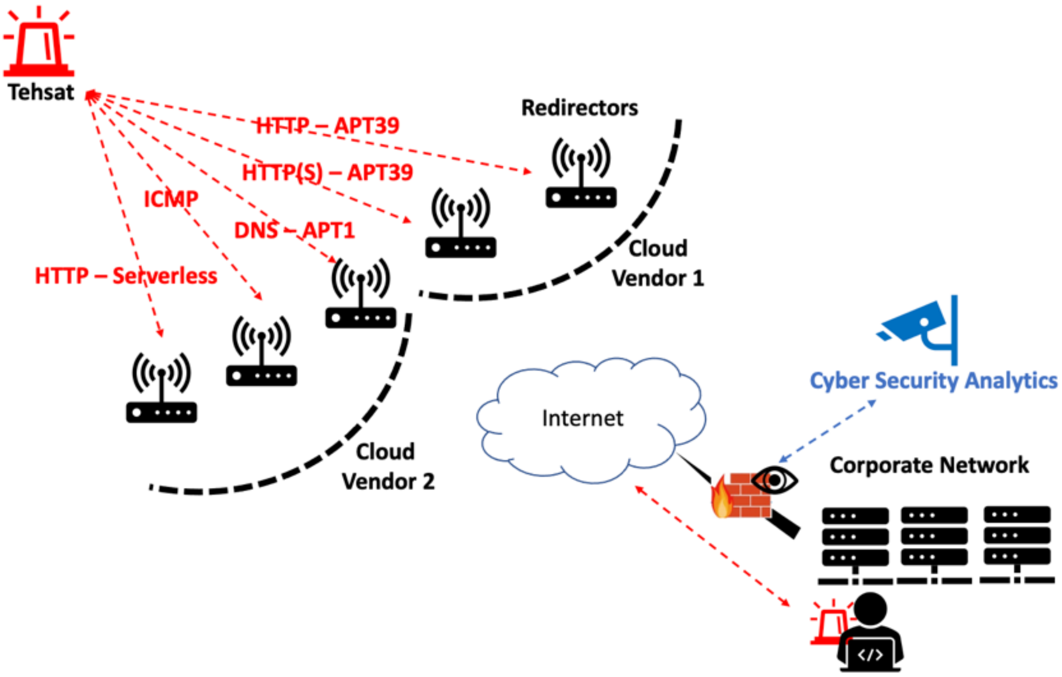


Figure 6 Tehsat – Distributed Deployment of Tehsat

Conclusion

Malware communications use various network protocols with resiliency improvements such as profiling, domain fronting and leveraging legitimate cloud services. Traditional security controls would be insufficient to detect customised C2 communications of last decade. Therefore, Cyber Security Analytics (CSA) is used for collecting related network and endpoint security control logs to achieve meaningful detections using correlations and deep learning. In academic research papers, adversary simulations are widely used to generate malware activities to verify efficiency of CSA detections. Due to the defence tool limitations, there is a gap for simulating multi-layered and distributed malware communications. In this paper, to mitigate this gap, I introduced Tehsat to generate malware communications in various protocols compatible with a scenario. Using this approach, it's possible to identify the gaps of CSA, analyse its efficiency against distributed C2 techniques, and improve CSA detections.

Appendices

Malware Communications and Indicators of Compromise

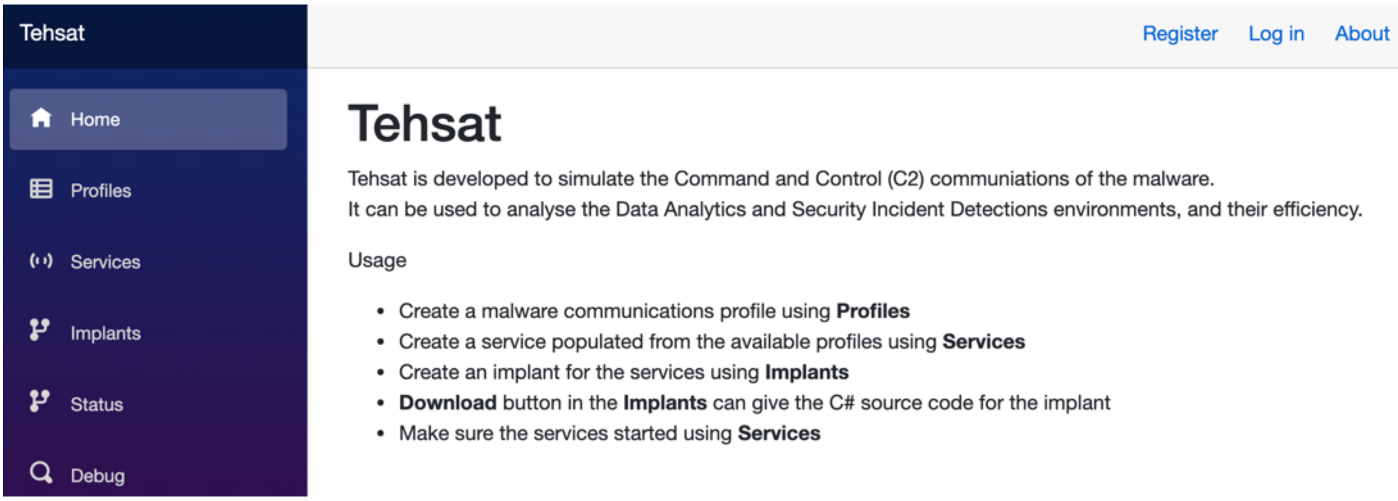
Protocol	Usage Examples	Common Behaviours	Indicators of Compromise
HTTP(S)	Mocking a legitimate web site or micro service API HTTP request/response for communications.	Regular check-ins Repeating requests Base64 content	Domain First Use Update Interval Constant Communications
DNS	Using DNS server responses for C2 discovery, malware registration or target verifications.	DNS SOA records A/CNAME resolutions Unusual Subdomains	Domain First Use Repeating DNS requests Update Interval
ICMP	Using ICMP data for C2 discovery, malware registration or target verifications.	ICMP Ping requests ICMP Error messages	ICMP over Internet Constant ICMP Requests Constant ICMP Errors Update Interval
TCP/UDP	Using established TCP/UDP sessions or half handshakes for malware communications.	Continuous communications Usually encrypted	Update Interval Long keep-alive TCP/UDP over Internet
Images	Steganography is widely used to deliver a stage of malware. Images can be found in legitimate web sites.	Data in image headers Adding data after image	Geometry Checks Data after EoF Large Data in Headers
Websocket **WebRTC** **SOCKS**	Real-time C2 communications or tunnelling for interactive attacks	Traffic tunnelling Interactive operations	Domain First Use Update Interval Constant Communications Long keep-alive
DNS over HTTPS	Using DNS server responses for full or partial C2 communications	Domain fronting A/CNAME resolutions Unusual Subdomains	Domain First Use DNS over HTTPS Use Update Interval
SMB	Named Pipes and DCE RPC services are used for implant-to-implant communications.	Named Pipe Use DCE RPC Service Use	Unusual Named Pipe Name Custom DCE RPC Service

A Sample Threat Modelling for Malware Communications

Affected Networks	Threat Description	Risk	Mitigations / Detections
Corporate to Internet Cloud to Internet	**Threat actor can use HTTP(S) to deploy or drive a malware** *HTTP(S) is widely used for all C2 communications. In addition, Domain Fronting or legitimate websites could be used as cover.*	**High** *Likelihood: H* *Impact: H*	New domain detection Proxy for all HTTP(s) TLS interception TLS SNI/Domain Mapping
Internet to Corporate Internet to Cloud	**Threat actor can deploy a web shell to a container app or a security appliance** *Web shells are widely used to hide malicious activities for border controller or web app compromise*	**High** *Likelihood: H* *Impact: H*	Appliance ingress filtering Web app integrity checks DMZ traffic monitoring
Corporate to Internet Cloud to Internet	**Threat actor can use DNS A queries to redirect to C2 servers** *DNS is used for malware registrations, C2 task delivery and C2 server discovery. Only a limited number of threat actors use DNS.*	**High** *Likelihood: M* *Impact: H*	New domain detection Internal DNS enforced DNS integrity monitoring
Corporate to Internet Cloud to Internet	**Threat actor may use a legitimate DNS service via HTTPS to hide C2 activities** *DNS over HTTPS is leveraging legitimate DNS services and encryption. Setup and use are difficult. Only a limited number of threat actors use DNS over HTTPS.*	**Medium** *Likelihood: L* *Impact: M*	New domain detection Proxy for all HTTP(s) TLS interception TLS SNI/Domain Mapping DNS over HTTPS Monitors
Cloud Internal (VPC)	**Threat actor can compromise internal containers or instances over HTTPS or TCP** Container compromises and sandbox escapes may require internal communications of containers. SSH would be also used for management or access.	**Medium** *Likelihood: L* *Impact: M*	Monitor container traffic Monitor instance traffic Web app integrity checks
Corporate Internal Cloud Internal	**Threat actor can use SMB Named Pipes, TCP or SSH to manage malware deployed internally** SMB Named Pipes and TCP are highly used by threat actors for internal communications. SSH is also used for Unix/Linux servers.	**Medium** *Likelihood: L* *Impact: M*	Sensors for each segment Traffic monitoring on GWs Monitor wireless traffic Endpoint Events: -Remote logon attempts -Monitor Windows NPs -Monitor Linux SSH
Corporate Internal Cloud Internal	**Threat actor can use PSExec, WMI, WinRM, registry or scheduled tasks for fileless communications** The techniques which can be used for lateral movement can be also used for internal communications	**Low** *Likelihood: L* *Impact: L*	Sensors for each segment Monitor SMB/SSH Endpoint Events: -Remote logon attempts -Monitor Windows NPs -Monitor Linux SSH

Tehsat - Malware Traffic Generator

Tehsat Main Screen



Tehsat

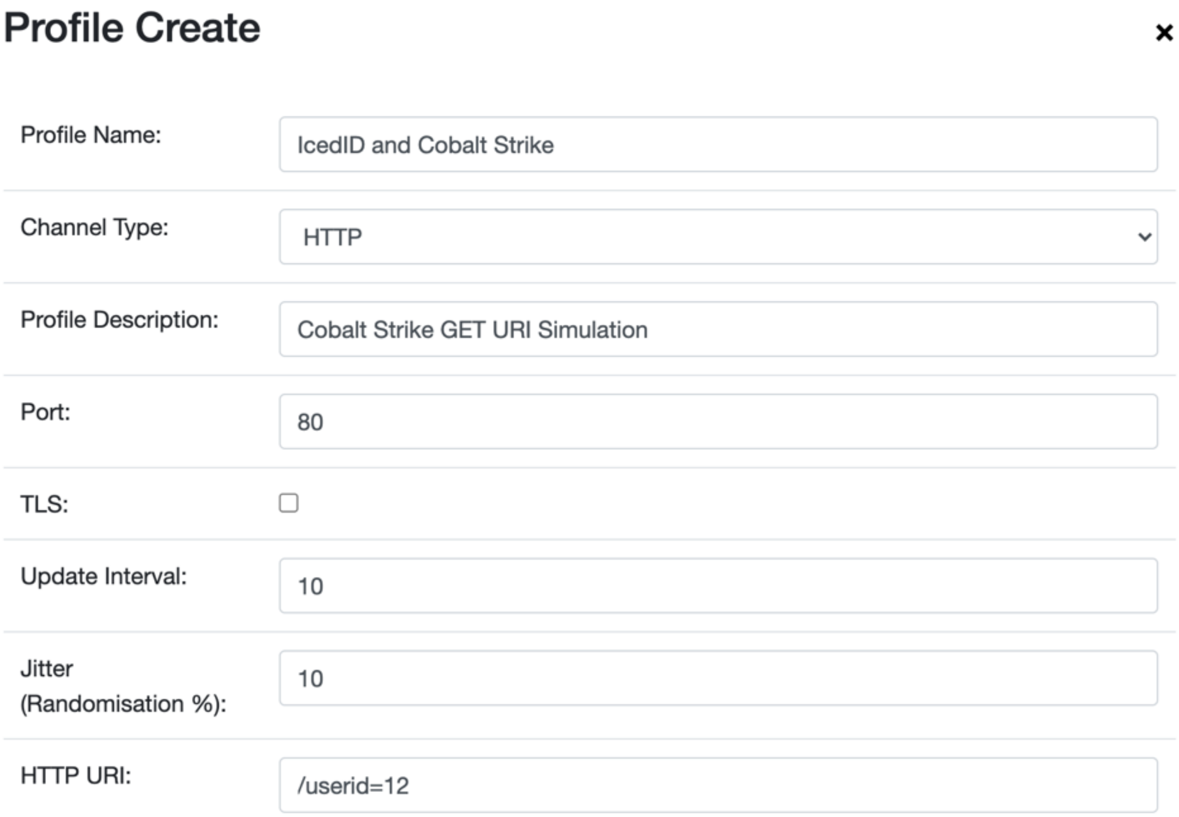
Tehsat is developed to simulate the Command and Control (C2) communiations of the malware.
It can be used to analyse the Data Analytics and Security Incident Detections environments, and their efficiency.

Usage

- Create a malware communications profile using **Profiles**
- Create a service populated from the available profiles using **Services**
- Create an implant for the services using **Implants**
- **Download** button in the **Implants** can give the C# source code for the implant
- Make sure the services started using **Services**

Creating Malware Communications Profile

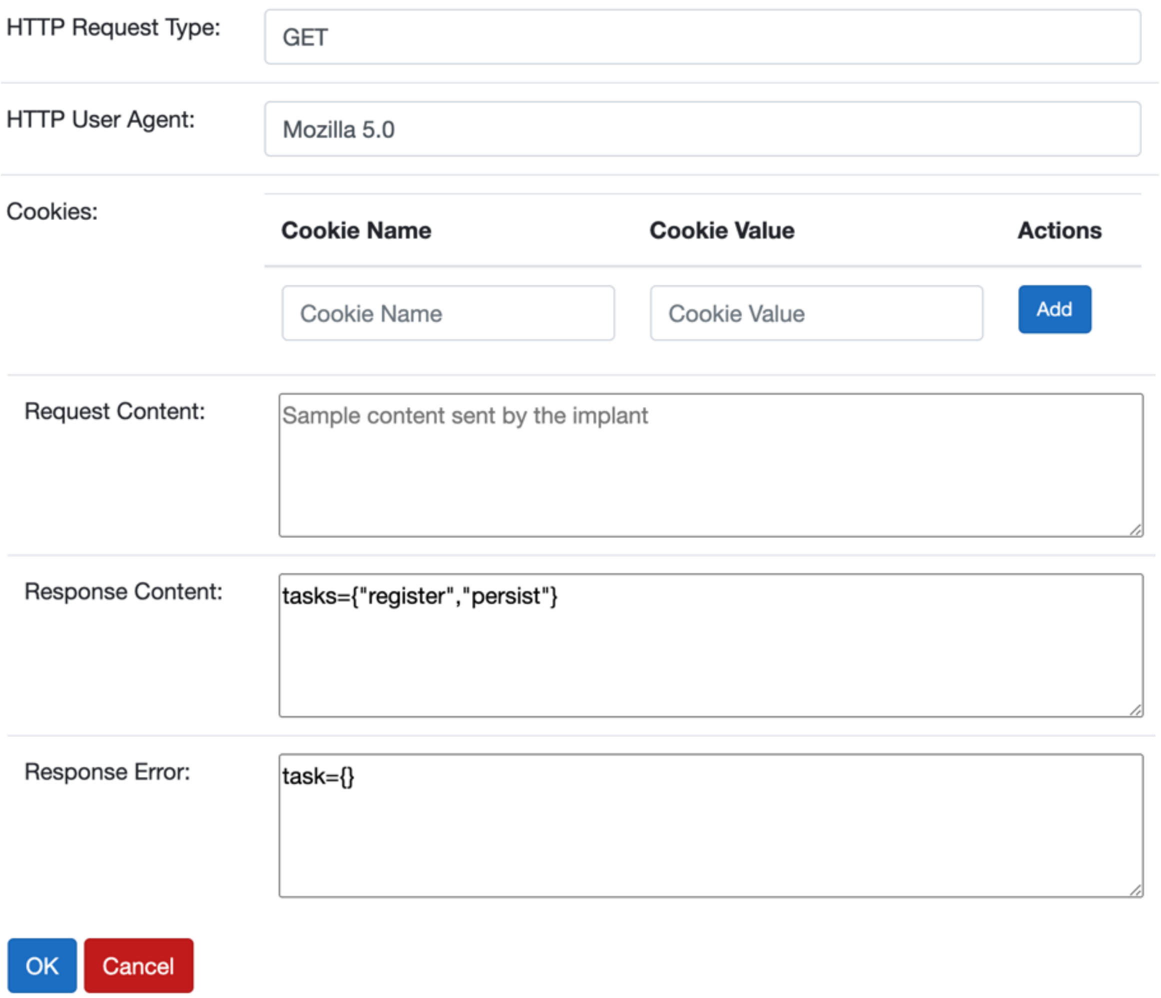
Tehsat profile create options such as protocol selection, HTTP options, HTTP headers and data to be simulated. I used **/userid=** URI given in E. Hjelmvik's blog post with **HTTP GET** method for this profile, and set update interval to 10ms with %10 randomisation for beaconing. GET doesn't require a data to post, so the request data is empty. The server response has some sample instructions if the URI matches with the given URI, otherwise sends no tasks. Tehsat can also add custom HTTP request and response headers, cookies, and user agents for the profiles.



Profile Create

Profile Name:	IcedID and Cobalt Strike
Channel Type:	HTTP
Profile Description:	Cobalt Strike GET URI Simulation
Port:	80
TLS:	□
Update Interval:	10
Jitter (Randomisation %):	10
HTTP URI:	/userid=12

Profile Name, Description and TCP/IP Options



HTTP Request Type:	GET
HTTP User Agent:	Mozilla 5.0

Cookies:

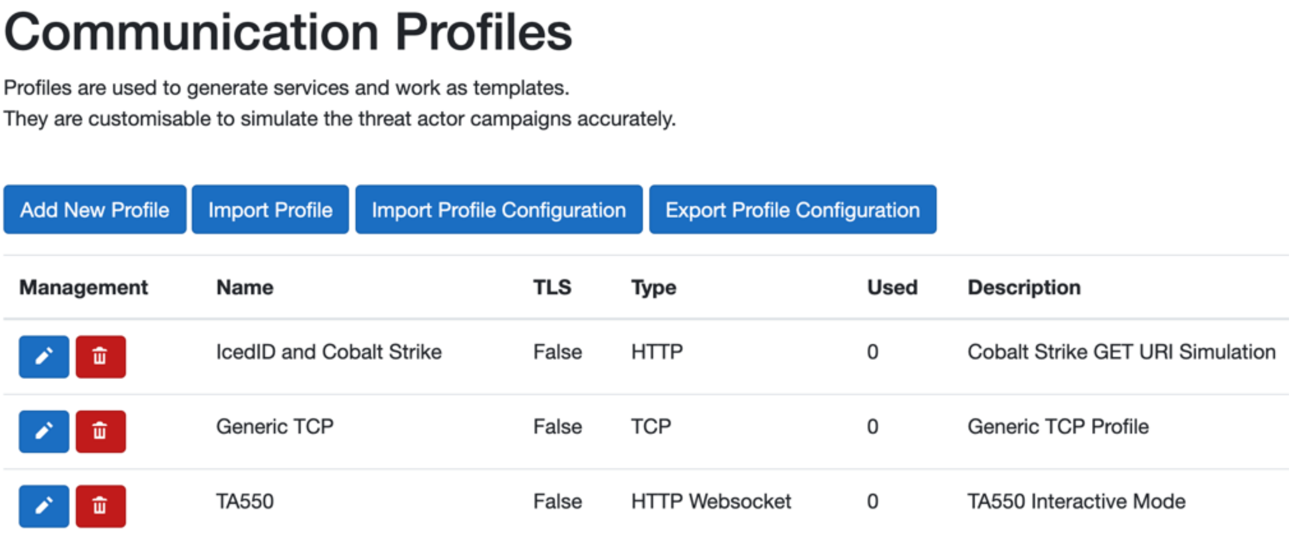
Cookie Name	Cookie Value	Actions
Cookie Name	Cookie Value	Add

Request Content:	Sample content sent by the implant
Response Content:	tasks={"register","persist"}
Response Error:	task={}

HTTP Options, Request and Response Types

Malware Communication Profiles and Services Available

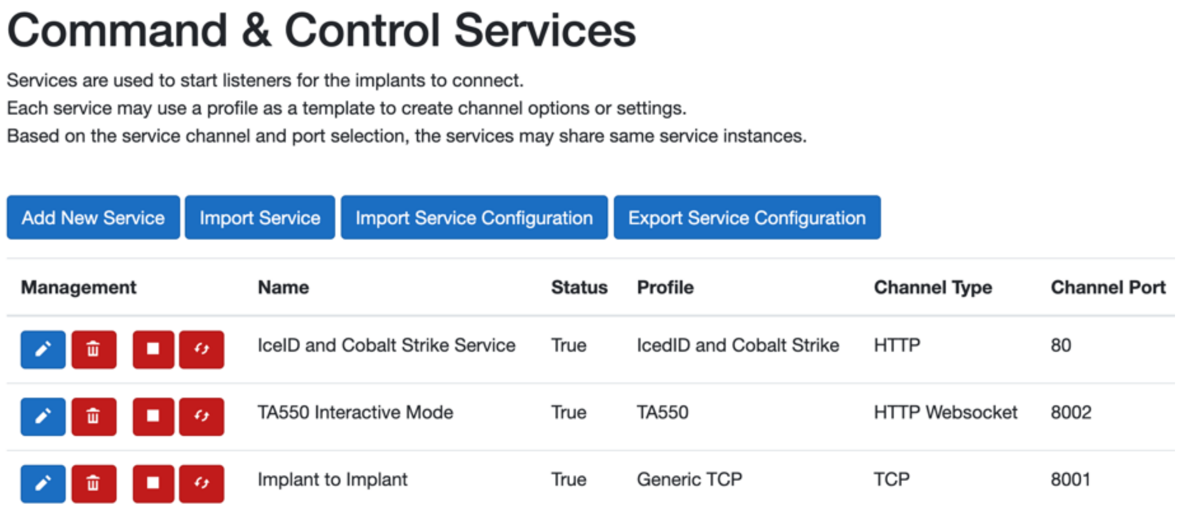
When the profile saved, it’s listed under the profiles section and ready to use in the service create as below. Using the base profile, multiple services can be created for different request types or service ports. This gives some flexibility to defenders to generate multiple communication types such as registration, task management and hands-on mode at the same time.



Communication Profiles

Profiles are used to generate services and work as templates.
They are customisable to simulate the threat actor campaigns accurately.

Management	Name	TLS	Type	Used	Description
✎ 🗑	IcedID and Cobalt Strike	False	HTTP	0	Cobalt Strike GET URI Simulation
✎ 🗑	Generic TCP	False	TCP	0	Generic TCP Profile
✎ 🗑	TA550	False	HTTP Websocket	0	TA550 Interactive Mode



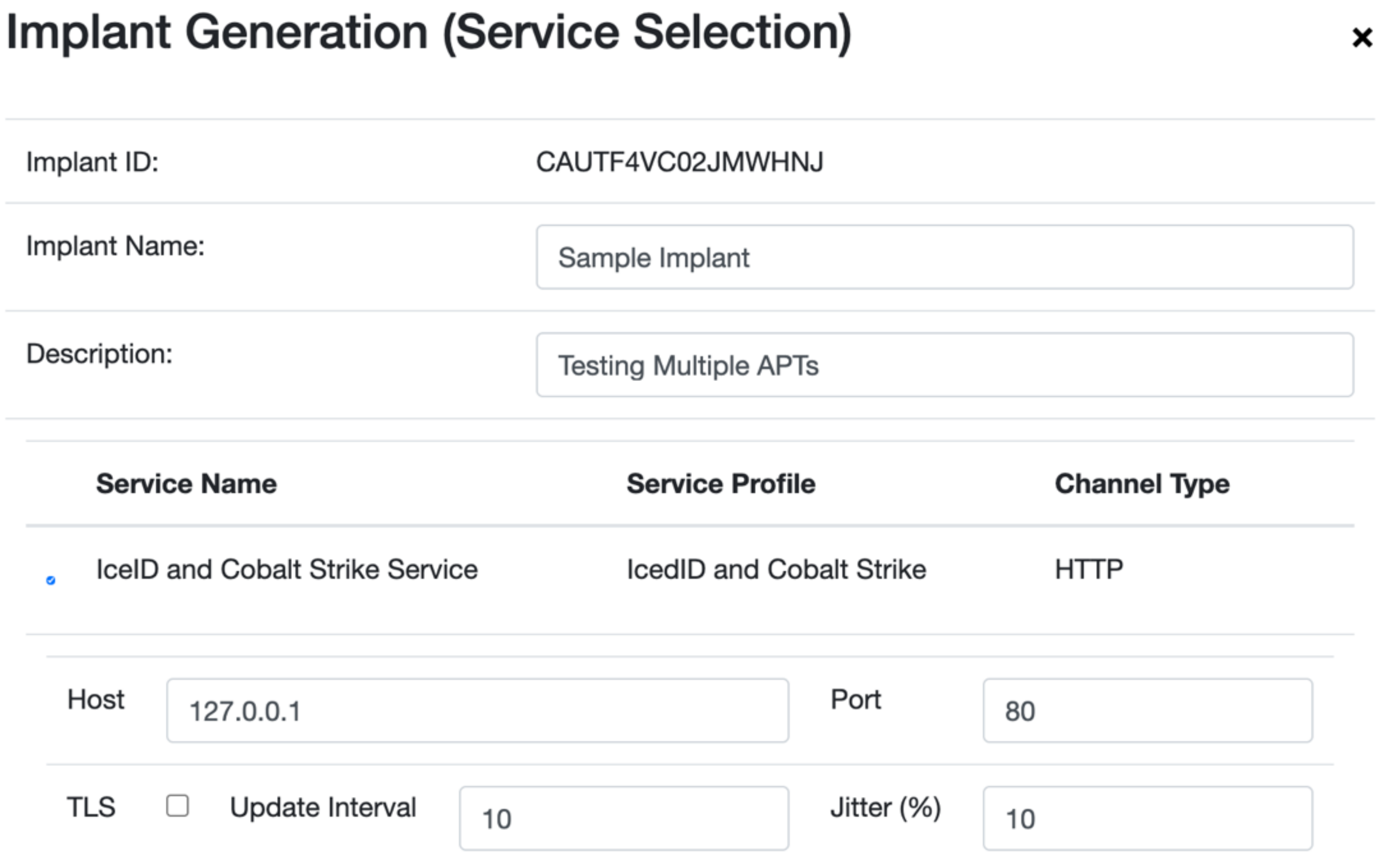
Command & Control Services
Add New Service
Import Service
Import Service Configuration
Export Service Configuration

Services are used to start listeners for the implants to connect.
Each service may use a profile as a template to create channel options or settings.
Based on the service channel and port selection, the services may share same service instances.

Management	Name	Status	Profile	Channel Type	Channel Port
	IceID and Cobalt Strike Service	True	IcedID and Cobalt Strike	HTTP	80
	TA550 Interactive Mode	True	TA550	HTTP Websocket	8002
	Implant to Implant	True	Generic TCP	TCP	8001

Implant Generation

The implant generation may use one or more C2 service selection to simplify the deployments. The implants can connect to the different domains and ports using different beaconing timings. The main use of this is giving enough flexibility to deploy multiple API redirectors around the Tehsat services on different domains, and sampling multiple malware activities at the same time. When the implant generated, it can be listed as in implants section, and can be downloaded as Visual Studio project or raw source code.

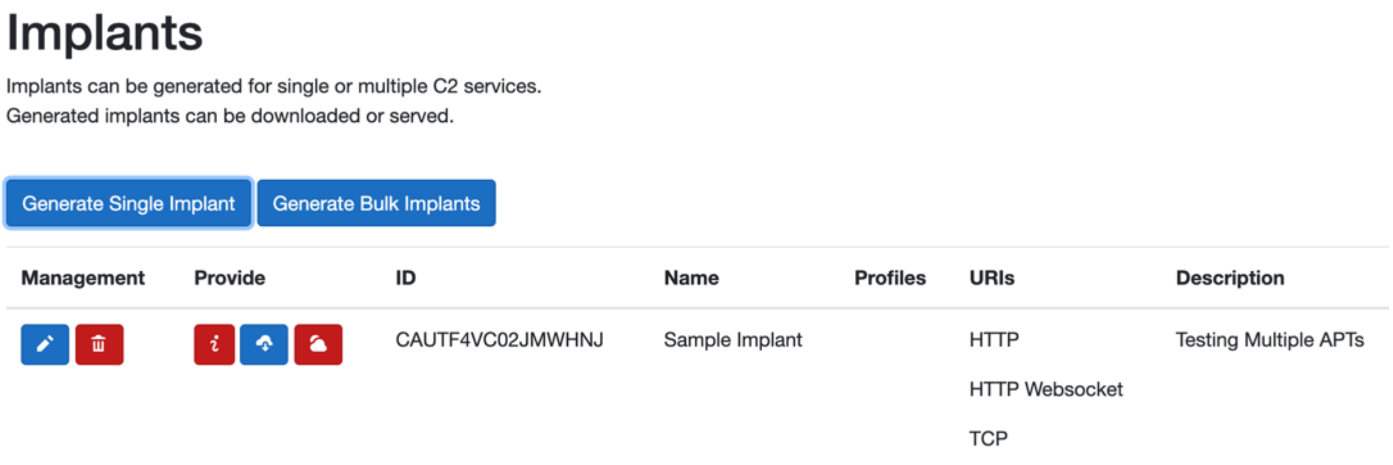


Implant ID:	CAUTF4VC02JMWHNJ
Implant Name:	Sample Implant
Description:	Testing Multiple APTs

	Service Name	Service Profile	Channel Type
●	IceID and Cobalt Strike Service	IcedID and Cobalt Strike	HTTP

Host	127.0.0.1	Port	80
TLS □ Update Interval	10	Jitter (%)	10

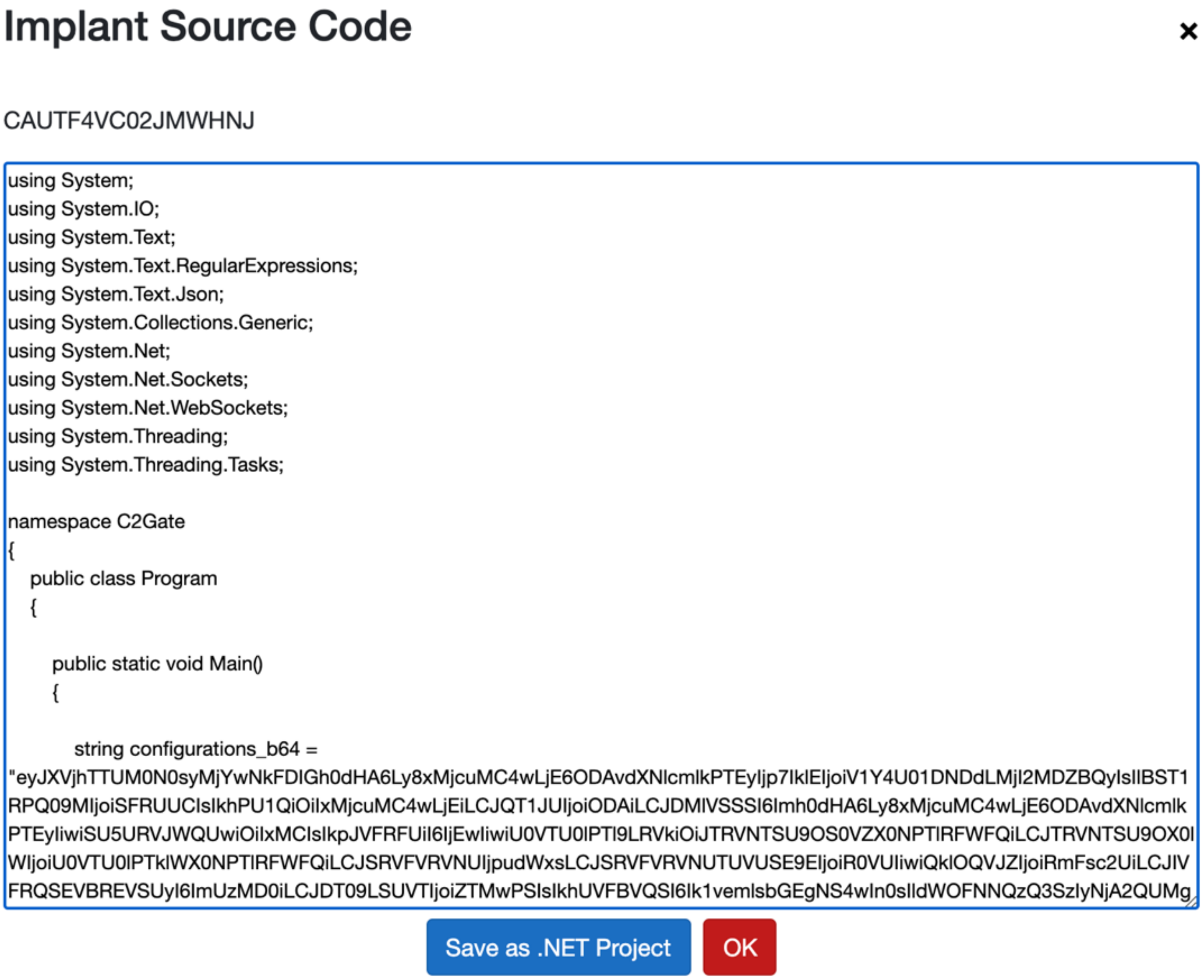
Implant and Communication Options



Implants

Implants can be generated for single or multiple C2 services.
Generated implants can be downloaded or served.

Management	Provide	ID	Name	Profiles	URIs	Description
✎ 🗑	ℹ ⬆ ☁	CAUTF4VC02JMWHNJ	Sample Implant		HTTP HTTP Websocket TCP	Testing Multiple APTs



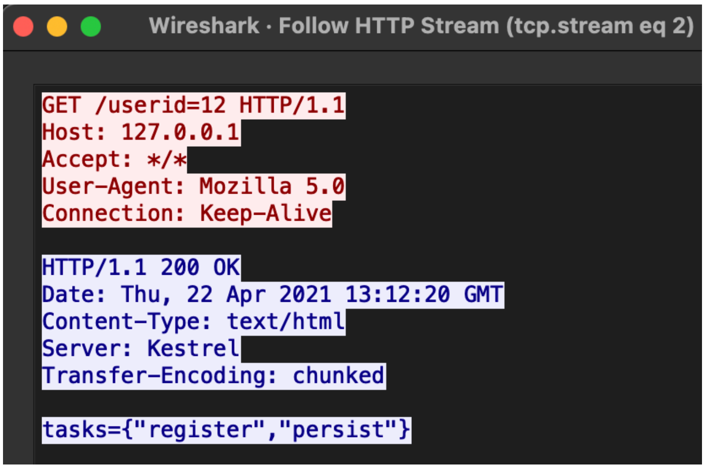
Implant Source Code

CAUTF4VC02JMWHNJ

```
using System;
using System.IO;
using System.Text;
using System.Text.RegularExpressions;
using System.Text.Json;
using System.Collections.Generic;
using System.Net;
using System.Net.Sockets;
using System.Net.WebSockets;
using System.Threading;
using System.Threading.Tasks;

namespace C2Gate
{
    public class Program
    {

        public static void Main()
        {

            string configurations_b64 =
"eyJXVjhTTUM0N0syMjYwNkFDIGh0dHA6Ly8xMjcuMC4wLjE6ODAvdXNlcmlkPTEyIjp7IklEIjoiV1Y4U01DNDdLMjI2MDZBQyIsIlBST1
RPQ09MIjoiSFRUUCIsIkhPU1QiOiIxMjcuMC4wLjEiLCJQT1JUIjoiODAiLCJDMlVSSSI6Imh0dHA6Ly8xMjcuMC4wLjE6ODAvdXNlcmlk
PTEyIiwiSU5URVJWQUwiOiIxMCIsIkpJVFRFUiI6IjEwIiwiU0VTU0lPTl9LRVkiOiJTRVNTSU9OS0VZX0NPTlRFWFQiLCJTRVNTSU9OX0l
WIjoiU0VTU0lPTklWX0NPTlRFWFQiLCJSRVFVRVNUIjpudWxsLCJSRVFVRVNUTUVUSE9EIjoiR0VUIiwiQklOQVJZIjoiRmFsc2UiLCJIV
FRQSEVBREVSUyI6ImUzMD0iLCJDT09LSUVTIjoiZTMwPSIsIkhUVFBVQSI6Ik1vemlsbGEgNS4wIn0sIldWOFNNQzQ3SzIyNjA2QUMg
```

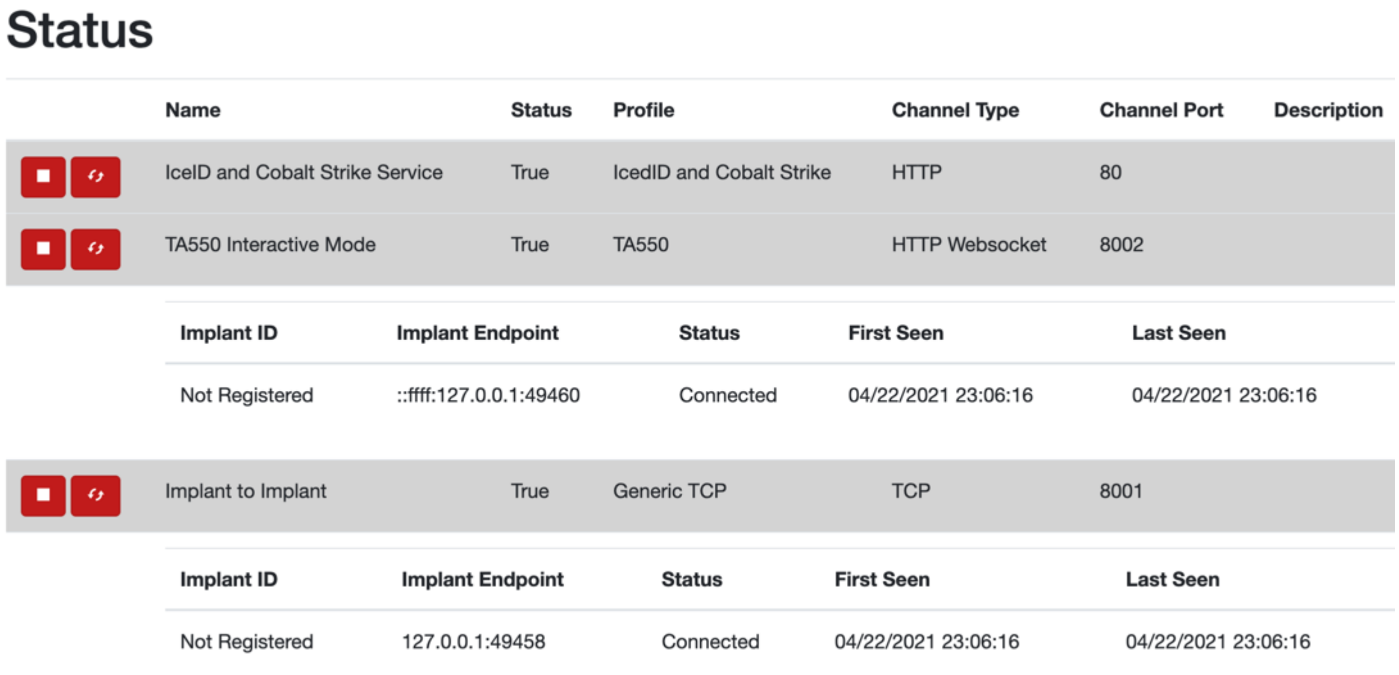
Monitoring Malware Communications

When the implant compiled and executed (dotnet run), the implant will connect to the C2 services designed and configured. The implant compiled can be deployed to the designated victims using PowerShell or group policies as it’s a .NET 5 assembly. The following capture shows the sample traffic generated for the first IcedID URI. So, the steps above can be repeated for each URI or service to generate different channels.



```
GET /userid=12 HTTP/1.1
Host: 127.0.0.1
Accept: */*
User-Agent: Mozilla 5.0
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Thu, 22 Apr 2021 13:12:20 GMT
Content-Type: text/html
Server: Kestrel
Transfer-Encoding: chunked

tasks={"register","persist"}
```

The service status and implant connection details are also available under the status section as below.



Status

	Name	Status	Profile	Channel Type	Channel Port	Description
■ ↻	IceID and Cobalt Strike Service	True	IcedID and Cobalt Strike	HTTP	80	
■ ↻	TA550 Interactive Mode	True	TA550	HTTP Websocket	8002	

Implant ID	Implant Endpoint	Status	First Seen	Last Seen
Not Registered	::ffff:127.0.0.1:49460	Connected	04/22/2021 23:06:16	04/22/2021 23:06:16

	Name	Status	Profile	Channel Type	Channel Port	Description
■ ↻	Implant to Implant	True	Generic TCP	TCP	8001	

Implant ID	Implant Endpoint	Status	First Seen	Last Seen
Not Registered	127.0.0.1:49458	Connected	04/22/2021 23:06:16	04/22/2021 23:06:16

References

[i] Fatih Ozavci September 2020. “*Current State of Malware Command and Control Communication Channels and Future Predictions*”. [Online]. Available: https://github.com/fozavci/ta505plus/blob/main/Current%20State%20of%20Malware%20Command%20and%20Control%20Channels%20and%20Future%20Predictions-v1.0.pdf [Accessed 18 April 2020].

[ii] ESET, "*Cyber espionage group, Turla, new campaign uses Instagram to spy on its targets*" June 2017. [Online]. Available: https://www.eset.com/us/about/newsroom/press-releases/cyber-espionage-group-turla-new-campaign-uses-instagram-to-spy-on-its-targets/. [Accessed 8 April 2021].

[iii] Kaspersky, "*Platinum is back*" June 2019. [Online]. Available: https://securelist.com/platinum-is-back/91135/ [Accessed 18 April 2020].

[iv] Blackberry Cylance, " *OceanLotus Steganography* " February 2019. [Online]. Available: https://www.blackberry.com/content/dam/bbcomv4/blackberry-com/en/company/research-and-intelligence/OceanLotus-Steganography-Malware-Analysis-White-Paper.pdf [Accessed 18 April 2020].

[v] ZDNet, "*Iranian hacker group becomes first known APT to weaponize DNS-over-HTTPS (DoH)*" August 2020. [Online]. Available: zdnet.com/article/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/ [Accessed 18 April 2020].

[vi] Fifield, D. et al 2015. Blocking-resistant communication through domain fronting. *Proceedings on Privacy Enhancing Technologies*, *2015*(2), 46-64.

[vii] Dutta, V. et al 2020. A deep learning ensemble for network anomaly and cyber-attack detection. *Sensors*, *20*(16), 4583.

[viii] Stergiopoulos, G. et al 2018. Automatic detection of various malicious traffic using side channel features on TCP packets. In *European Symposium on Research in Computer Security* (pp. 346-362). Springer, Cham.

[ix] The MITRE Corporation, "*Adversarial Tactics, Techniques and Common Knowledge*," April 2021. [Online]. Available: https://attack.mitre.org/. [Accessed 8 April 2020].

[x] Strategic Cyber LLC, a HelpSystems company, "*Cobalt Strike*". [Online]. Available: https://www.cobaltstrike.com/ [Accessed 10 April 2021].

[xi] Merlin, "*Merlin Command and Control Server*". [Online]. Available: https://github.com/Ne0nd0g/merlin [Accessed 10 April 2021].

[xii] Godoh, " *A DNS-over-HTTPS Command & Control Proof of Concept*". [Online]. Available: https://github.com/sensepost/godoh [Accessed 10 April 2021].

[xiii] Rapid 7, "*Metasploit Framework*". [Online]. Available: https://www.metasploit.com/ [Accessed 10 April 2021].

[xiv] "PowerShell Empire". [Online]. Available: https://www.powershellempire.com/ [Accessed 10 April 2021].

[xv] Strom, B. et al 2017. Finding cyber threats with ATT&CK-based analytics. *The MITRE Corporation, Bedford, MA, Technical Report No. MTR170202*.

[xvi] Applebaum, A. et al 2016. Intelligent, automated red team emulation. In *Proceedings of the 32nd Annual Conference on Computer Security Applications* (pp. 363-373).

[xvii] Musman, S. et al 2019. Steps toward a principled approach to automating cyber responses. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications* (Vol. 11006, p. 110061E). International Society for Optics and Photonics.

[xviii] Applebaum, A. et al 2017. Analysis of automated adversary emulation techniques. In *Proceedings of the Summer Simulation Multi-Conference* (pp. 1-12).

[xix] Miller, D. et al 2018. Automated adversary emulation: A case for planning and acting with unknowns. *MITRE: McLean, VA, USA*.

[xx] The MITRE CALDERA, April 2021. [Online]. Available: https://github.com/mitre/caldera. [Accessed 8 April 2021].

[xxi] Fatih Ozavci, "*Petaq C2 – Purple Team C2 and Malware*". [Online]. Available: https://github.com/fozavci/petaqc2/ [Accessed 10 April 2021].

[xxii] Fatih Ozavci, "*TA505+ Adversary Simulation Pack*". [Online]. Available: https://github.com/fozavci/ta505plus [Accessed 10 April 2021].

[xxiii] Fatih Ozavci, "*Projecting TA505 Tradecraft to Cutting Edge Systems*". [Online]. Available: https://www.youtube.com/watch?v=ymp8diKJBRk&ab_channel=SCYTHE [Accessed 10 April 2021].

[xxiv] Carvalho, M. et al 2013. MTC2: A command and control framework for moving target defense and cyber resilience. In *2013 6th International Symposium on Resilient Control Systems (ISRCS)* (pp. 175-180). IEEE.

[xxv] Erik Hjelmvik, Security Boulevard, "Analysing a malware PCAP with IcedID and Cobalt Strike traffic". [Online]. Available: https://securityboulevard.com/2021/04/analysing-a-malware-pcap-with-icedid-and-cobalt-strike-traffic/ [Accessed 20 April 2021].