ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

Development of Command and Control Server (C2 Server)

Aman Dekate[1]; Dev Mulchandani[2]; Dr. Sampada Wazalwar[3]; Chinmay Rahangdale[4]; Gaurav Choudhari[5]; Swati Tiwari[6]

[1,2,3,4,5,6]Center of Excellence (Information Security), G H Raisoni College of Engineering Nagpur,

Publication Date: 2025/05/13

Abstract: This research paper delves into the problematic architecture of HTTP/HTTPS-based Command and Control (C2) servers, a pivotal aspect in present day cyberattacks. We look at the strategies hired with the aid of C2 servers to set up covert communication channels, evade detection, and keep control over compromised systems. The paper explores the function of cloud-primarily based infrastructure in improving the scalability and resilience of C2 servers, while also discussing the challenges it poses for cybersecurity specialists. By understanding the mechanisms and strategies hired by way of C2 servers, we aim to make contributions to the development of extra effective defense mechanisms and mitigate the impact of cyber threats.

How to Cite: Aman Dekate; Dev Mulchandani; Dr. Sampada Wazalwar; Chinmay Rahangdale; Gaurav Choudhari; Swati Tiwari. (2025). Development of Command and Control Server (C2 Server). *International Journal of Innovative Science and Research Technology*, 10(4), 3456-3462. https://doi.org/10.38124/ijisrt/25apr1891.

I. INTRODUCTION

The stopping point of a network system is related to its complexity, functionality and connectivity[11]. The enemy and its tactics, methods and systems have become more complex, better funded and able to run on a fast machine[12]. The impact of a cyber attack can affect the health of a country and the longevity of an organization, even if it results in significant financial losses and intellectual property loss[6], or more serious disruptions such as the rapid failure of a key country[2]. It has caused great chaos in society and even loss of life.

Cyber defense systems often operate independently and are often statically configured, which can lead to network security operations failing[2]. Therefore, detecting and responding to cybersecurity incidents, especially large and complex incidents, is a continuous and difficult process, presenting an asymmetric opportunity for adversaries to succeed and control the fight[7].

- *Open Command and Control - OpenC2:*

Open Command and Control (OpenC2) is a language designed to improve coordination and collaboration across cyber defense technologies[13]. By providing a unified framework for issuing and responding to commands, OpenC2 enables disparate security tools and systems to work together seamlessly[1]. This design is critical for improving the efficiency and effectiveness of cybersecurity operations, resulting in faster and more coordinated responses to threats[4]. By simplifying automation and orchestration, OpenC2 helps organizations improve their defenses, shorten response times, and increase overall cyber resilience[14].

II. DETAILS OF C2 SERVER

A. Functionality:

- *Centralized Control:*

The C2 server acts as a central control point for attackers, allowing them to issue commands to compromised systems and receive information from them[12].

Manages the infected individuals' work, enabling collaboration and cooperation.

- *Command Distribution:*

Attackers use C2 servers to send commands to viruses. These instructions may include tasks such as deleting files[11], downloading and running additional malware, or additional attacks.

Commands are often encrypted or obfuscated to avoid detection by security mechanisms[7].

- *Data Exfiltration:*

An infected host can send the log back to the C2 server[9]. This information may contain sensitive information such as access credentials, financial information, or intellectual property rights.

C2 servers collect and organize this information for attackers to use[9].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

➢ *Update and Maintenance:*

C2 servers push updates to the malware on the affected system, allowing the malware to remain active and undetected[4].

This includes implementing new features, fixing bugs, or modifying the command line to avoid detection[5].

➢ *Stealth and Evasion:*

C2 servers often use a variety of techniques to avoid detection and compromise by security devices. This includes encryption, domain streaming, high-speed connections, and peer-to-peer (P2P) communications[8].

Some advanced C2 architectures use structural design to strengthen attack resistance[3].

➢ *Persistence:*

C2 servers help keep malware at bay. They can instruct malware to reinstall or take steps to prevent its removal[6].

They can also help malware adapt to changes in the target's environment, such as updates to the host operating system or security software[14].

B. Types of C2 Server:

➢ *HTTP/HTTPS C2 Servers*

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are widely utilized protocols for command and control (C2) communication in cybersecurity contexts[3].

HTTP C2 Server: HTTP is the basic protocol for communication on the web and facilitates the transmission of commands between C2 servers and infected individuals[1]. In this configuration, the infected client sends regular HTTP requests to the C2 server, which then responds with commands or updated information[6]. This approach leverages network-wide location to prevent malicious activity and makes it difficult for network monitoring tools to detect it[12].

HTTPS C2 Server: HTTPS connects to HTTP using SSL/TLS encryption, securing data transfer between the client and server[15]. This encryption not only protects the integrity and confidentiality of the data, but also adds a layer of obfuscation that makes it difficult for cybersecurity systems to analyze traffic[13]. HTTPS is particularly effective at hiding C2 communications due to its encrypted properties, but it can also cause concern if malicious traffic patterns are detected[7].

- *Advantages:*

HTTP/HTTPS: The web-wide environment allows this process to integrate with legitimate web activity, helping to avoid detection. HTTPS further enhances privacy through encryption[8].

- *Challenges:*

HTTP/HTTPS: The volume and pattern of network connections can sometimes be suspicious. Advanced detection techniques can flag suspicious patterns or behaviors, even in camouflage[9].

➢ *DNS-Based C2 Servers*

DNS (Domain Name System) is another protocol used for C2 communication, which uses the DNS query and response system to control the system[10].

- DNS C2 server: In a DNS-based C2 configuration, infected individuals communicate with the C2 server via DNS queries[6]. These queries typically include commands or information in DNS requests or responses (for example, placing information in a subdomain)[13]. The legal nature of the DNS protocol and its ability to bypass network security often make it a good choice for encrypted communications[12].

- *Advantages:*

✓ DNS: The protocol's central role in the operation of the Internet allows it to bypass many security measures, making it a good way to evade detection. DNS traffic is generally subject to less scrutiny than other types of web traffic[8].

- *Challenges:*

✓ DNS: The efficiency of DNS for large data transfers is limited compared to HTTP/HTTPS[7]. Excessive or unusual DNS activity can also raise red flags in security monitoring systems, potentially revealing the presence of a C2 infrastructure[8].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

III. ARCHITECTURE

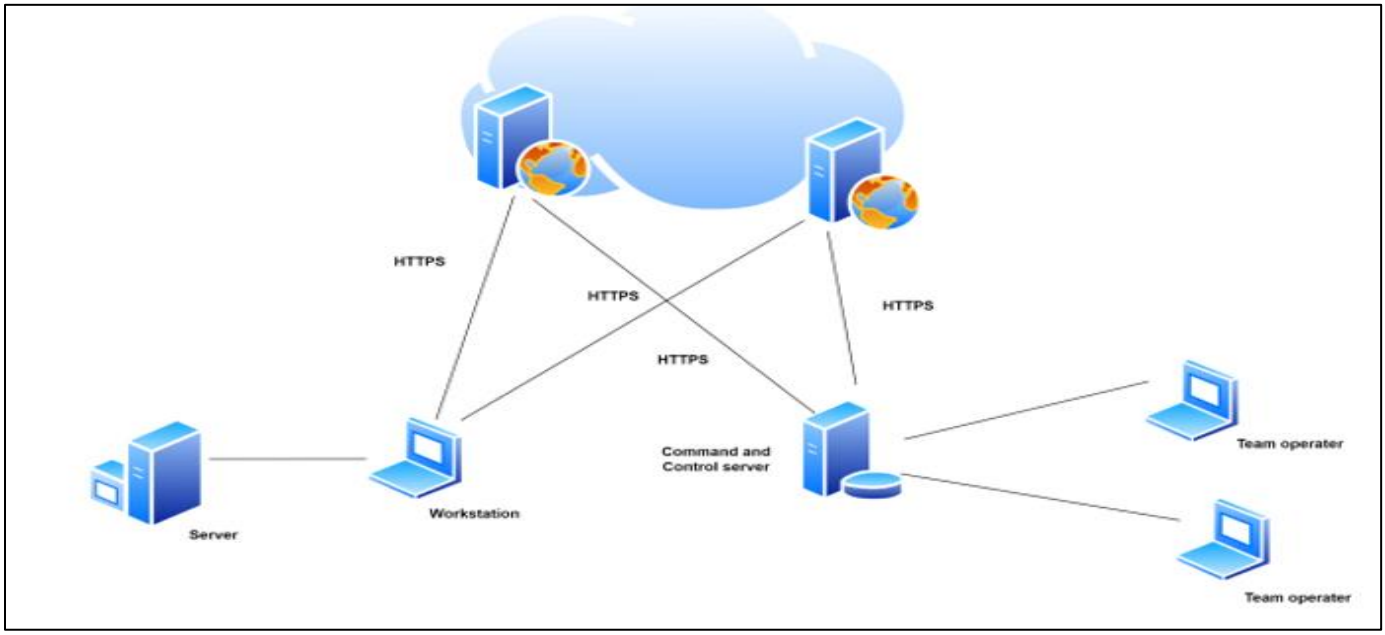

Fig : C2 Server Network Framework

An HTTP/HTTPS-based Command and Control (C2) server leverages the ubiquity of web traffic to establish covert communication channels with compromised systems (bots)[12]. This architecture typically follows a client-server model.

A. *Core Components:*

➢ *C2 Server[6]:*

- Listens on a specific port (usually 80 for HTTP or 443 for HTTPS).
- Handles incoming connections from compromised systems (bots).
- Processes incoming requests, extracting commands or data.
- Sends responses (commands or data) to bots.
- May employ a database to store bot information, command history, and collected data.

➢ *Bots (Compromised Systems)[7]:*

- Initiate connections to the C2 server.
- Send data (e.g., system information, stolen data) to the server.
- Receive commands from the server.
- Execute commands and return results.

B. *Communication:*

- HTTP/HTTPS Requests: Bots communicate with the C2 server by sending HTTP/HTTPS requests[6].
- Custom Protocols: To evade detection, C2 servers often employ custom protocols embedded within HTTP/HTTPS requests and responses[9]. This involves defining specific data formats, headers, or parameters for command and control communication[9].
- Data Exfiltration**:** Stolen data is typically exfiltrated through HTTP/HTTPS requests to the C2 server[6].

C. *Security Considerations:*

- Obfuscation: C2 servers employ various techniques to hide malicious activity within legitimate HTTP/HTTPS traffic. This can include custom headers, unusual URL structures, or encrypted data[8].
- Encryption: To protect sensitive data, C2 servers often use encryption to secure communication[7].
- Anti-Analysis: To hinder security researchers, C2 servers may implement techniques like code obfuscation, packing, or virtualization[]14.

IV. WORKING

➢ *How an HTTP/HTTPS Based C2 Server Works?*

An HTTP/HTTPS-based C2 server operates as a centralized command and control hub, communicating with compromised systems (bots) through the familiar HTTP/HTTPS protocol[15].

➢ *Operational Breakdown:*

- Server Setup: The C2 server is configured to listen on standard HTTP/HTTPS ports (80 and 443) for incoming connections. It typically employs a web server (like Apache or Nginx) to handle incoming requests[11].
- Bot Connection**:** Compromised systems (bots) initiate connections to the C2 server using standard HTTP/HTTPS requests[6]. This can be achieved through various methods, including direct connections, domain fronting, or other obfuscation techniques[9].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

- Data Exchange: Communication between the C2 server and bots occurs through HTTP/HTTPS requests and responses[8]. To conceal malicious activity, C2 servers often employ custom protocols embedded within these requests[13]. This involves defining specific data formats, headers, or parameters for command and control communication[4]. Commands, data, and control information are typically encoded within the request or response body[3].
- Command and Control: The C2 server sends commands to bots embedded within the HTTP/HTTPS traffic[5]. These commands can range from simple data exfiltration to complex actions like downloading additional malware or launching attacks[5]. Bots process these commands and execute them accordingly[2].
- Data Exfiltration: Stolen data is exfiltrated from compromised systems to the C2 server using HTTP/HTTPS requests. Data is often encoded or encrypted to evade detection[2].
- Persistence:C2 servers implement mechanisms to maintain persistent connections or establish reconnections to ensure continuous communication with bots[5]. This can involve techniques like keep- alive connections, heartbeat packets, or domain generation algorithms (DGAs) to generate new domain names[13].

V. ATTACK AND DEFENSE MECHANISM

A. *Attacking Method*

Displaying how an attack takes place from a remote server with a payload which can be uploaded to the victim host via social engineering means[11].

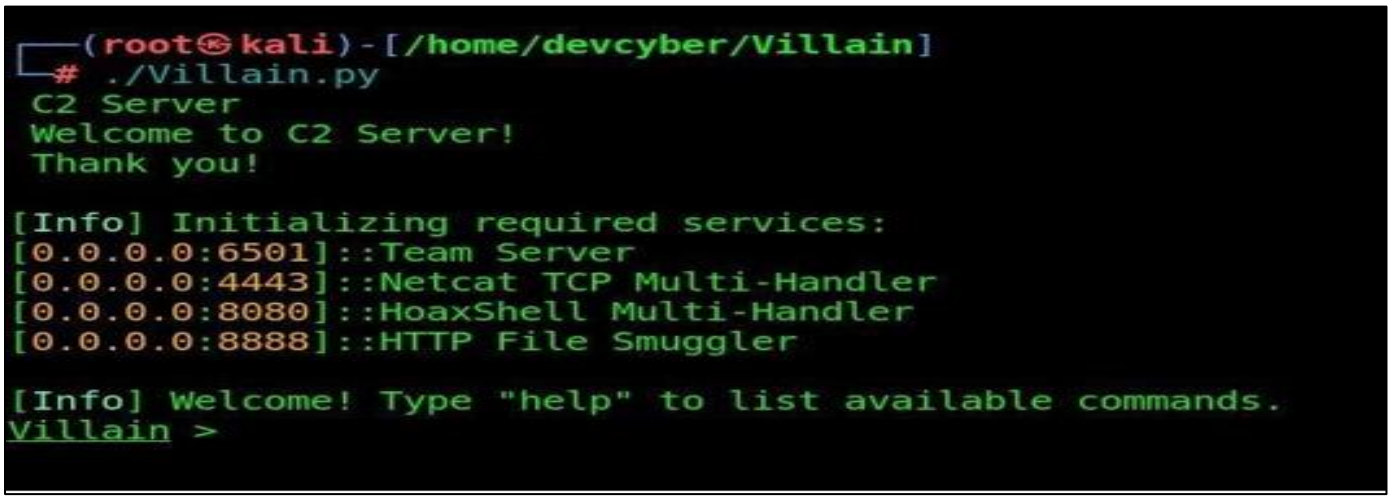

Fig 2: Initialized C2 Server

Initializing our C2 server, with our attack surface setup on localhost on port 6501. This is the host network that is initiated with our python code. We also get a default 'help' command which displays various commands present on our server[10].

Fig 3: Commands in C2 Server

These are the list of commands that are available on the C2 server[10].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

Fig 4: Malicious Payload Generation

This is a payload generated to take over a windows host. This payload needs to be executed on the victim host to take its control[4].

Fig 5: Backdoor Session Established

After the payload was executed we can see that a backdoor session was established. Now, we can access the shell as the victim user from the server's shell[15].

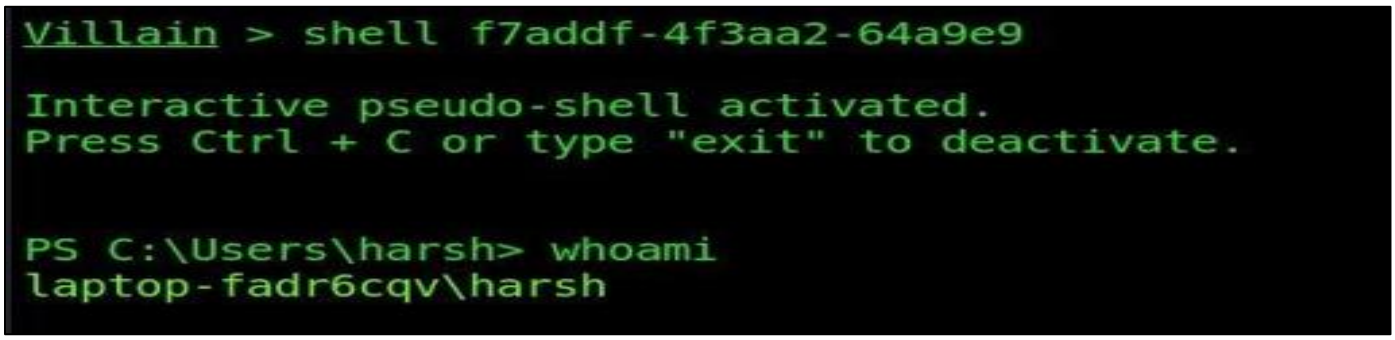

Fig 6: Shell Takeover of Victim's Device We Have Access Over the Victim's Shell[11].

Fig 7: Display of Victim's System Files We Can Access the Files of the Victim User[8].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

B. Defensive Method

➢ *Network Security[8]*

- Firewall Rules: Implement strict firewall rules to control inbound and outbound traffic. Only allow connections from trusted IP addresses and ports.
- Network Segmentation: Isolate your C2 server in a separate network segment from other critical systems to limit the impact of a potential breach.

➢ *Authentication and Access Control[9]*

- Strong Authentication: Use multi-factor authentication (MFA) for access to the C2 server and administrative functions.
- Least Privilege Principle: Grant minimal permissions to users and services. Ensure that users have only the access they need to perform their tasks.

➢ *Data Protection[5]*

- Encryption: Use strong encryption for data at rest and in transit. Ensure that all communications with the C2 server are encrypted using protocols like TLS/SSL.
- Regular Backups: Maintain regular backups of critical data and configuration files. Ensure backups are stored securely and tested for restoration.

➢ *Intrusion Detection and Prevention[7]*

- Intrusion Detection Systems (IDS): Deploy IDS to monitor network traffic and detect suspicious activities or potential attacks.
- Intrusion Prevention Systems (IPS): Use IPS to actively block detected threats and prevent malicious activities.

VI. CONCLUSION

Command and control infrastructure detection is an increasingly important area of study. With threat actors continuously coming up with new techniques for hiding malware communications, security researchers are required to keep up and develop detection methods. In this work we have explored how to set up and establish a C2 server with a detailed study of its working with its agents and listeners. Moreover, we saw how a session is established via malicious payload which is executed on the victim's machine which can be transferred using social engineering means in order to take over remote access to the victim's machine. We have also discussed various means through which a defense could be validated on the victim's side.

REFERENCES

[1]. X. Guo, G. Cheng, Y. Hu and M. Dai, "Progress in Command and Control Server Finding Schemes of Botnet," 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 2016, pp. 1723-1727, doi: 10.1109/TrustCom.2016.0264.

[2]. D. Jakovljevic, J. Balen and K. Vidović, "Integration of traffic and travel data exchange in command and control platform," 2016 International Conference on Smart Systems and Technologies (SST), Osijek, Croatia, 2016, pp. 281-286, doi: 10.1109/SST.2016.7765674.

[3]. R. Pitsko and D. Verma, "System of System case study - why the network centric command and control system needed to change: Lessons from army command and control in Operation Iraqi Freedom," 2011 6th International Conference on System of Systems Engineering, Albuquerque, NM, USA,2011,pp.335-340, doi:10.1109/SYSOSE.2011.5966620.

[4]. E. Răduca, L. Nistor, C. Hatiegan, M. Răduca, I. Pădureanu and S. Drăghici, "Web server for command, control and monitoring of industrial equipment," 2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 2015, pp. 61-66, doi: 10.1109/ATEE.2015.7133673.

[5]. F. F. Etemad and P. Vahdani, "Real-time Botnet command and control characterization at the host level," 6th International Symposium on Telecommunications (IST), Tehran, Iran, 2012, pp. 1005-1009, doi: 10.1109/ISTEL.2012.6483133.

[6]. Y. Chen, J. Wang, Y. Zhang, W. Cai and Y. Li, "Multitasking Command and Control System of Equipment Test Based on Virtual Machine Platform," 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 2020, pp. 972-976, doi:10.1109/ICCCS49078.2020.9118496.

[7]. R. Eltomy and W. Lalouani, "Explainable Intrusion Detection in Industrial Control Systems," 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS), St. Louis, MO, USA, 2024, pp. 1-8, doi: 10.1109/ICPS59941.2024.10640024.

[8]. F. Mira, "An investigation of malware and the systems used to detect and identify malware," 2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP), Sousse, Tunisia, 2024, pp. 1-8, doi:10.1109/ATSIP62566.2024.10638878.

[9]. G. Grieco, D. Striccoli, G. Piro, R. Bolla, G. Boggia and L. A. Grieco, "Authentication and Authorization in Cyber-Security Frameworks: a Novel Approach for Securing Digital Service Chains," *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, Milan, Italy, 2022, pp. 468-473, doi: 10.1109/NetSoft54395.2022.9844030.

[10]. Sunoj and B. V. Sherif, "Varying Encryption Scheme: An Innovative Approach to Legacy Data Security," 2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS), Kanjirapally, India, 2023, pp. 1-5, doi: 10.1109/AICERA/ICIS59538.2023.10420200

[11]. Haider, R.Z., Aslam, B., Abbas, H. *et al.*C2-Eye:framework for detecting command and control

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1891

(C2) connection of supply chain attacks. *Int. J. Inf. Secur.* 23, 2531–2545 (2024).

[12]. F. Sadique and S. Sengupta, "Analysis of Attacker Behavior in Compromised Hosts During Command and Control, "ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, 2021, pp. 1-7, doi: 10.1109/ ICC42927.2021.9500859.

[13]. F. Dang, L. Yan and Y. Yang, "Research on Intelligent Centralized System Based on Security Architecture of Computer Cloud Security Protection," 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2023, pp. 1281-1285, doi: 10.1109/ICETCI57876.2023.10176977.

[14]. N.Kaur and M. Singh, "Botnet and botnet detection techniques in cyber realm, " 2016 International Conference on Inventive Computation Technologies (ICTCT), Coimbatore, India, 2016, pp. 1-7, doi:10.1109/INVENTIVE.2016.7830080.

[15]. S. Ramezany, R. Setthawong and T. Tanprasert, "A Machine Learning-based Malicious Payload Detection and Classification Framework for New Web Attacks," 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology(ECTI-CON), Prachuap Khiri, Thailand, 2022, pp. 1-4, doi: 10.1109/ECTI-CON54298.2022.9795455.