How to: Detect and prevent common data exfiltration attacks

blog.apnic.net/2022/03/31/how-to-detect-and-prevent-common-data-exfiltration-attacks/

Debashis Pal

March 30, 2022

Photo: No Access by Bob Shand, Flickr

Data exfiltration is a technique used by malicious actors to carry out an unauthorized data transfer from a computer resource. Data exfiltration can be done remotely or locally and can be difficult to detect from normal network traffic.

Types of data that are targeted include: Usernames, associated passwords and other system authentication-related information, cryptographic keys, financial records, information associated with strategic decisions, and mailing addresses with content or what is valuable data for a cyber attacker. The damages can immeasurable when the organization's most valuable data is in the hand of a cyber attacker.

Advanced Persistent Threats (APTs) are one form of cyberattack in which data exfiltration is often a primary goal. The goal of an APT is to gain access to a network, but remain undetected as it stealthily seeks out the most valuable or target data.

Usually, attackers use covert channels for data exfiltration because covert channels are usually very difficult to detect due to their ability to use existing legitimate connections or protocols. A covert channel is a method of data transfer between two parties (usually a malicious insider and a malicious outsider) over a medium that is not supposed to be allowed to communicate by the computer security policy. The Trusted Computer System Evaluation Criteria (TCSEC) defines two kinds of covert channels:

- Storage channels, which communicate by modifying a 'storage location', such as a hard drive.
- Timing channels, which perform operations that affect the 'real response time observed' by the receiver.

Unfortunately, an attacker does not need to use advanced tools to exfiltrate data. They can use very simple techniques for stealing and transferring data from an internal network to an attacker domain such as HTTP/HTTPS, SMTP, DNS, SMTP, P2P, VPN or even the ARP method for data exfiltration. For example, the MITRE ATT&CK framework exfiltration tactic (TA0010) describes how an attacker can take data collected within a target network and exfiltrate it outside the network to systems under the attacker's control.

In this post, I will review a few common data exfiltration techniques in a lab environment. I will also highlight the best practices for detecting and preventing data exfiltration attacks.

Note: All the cases in this post were tested in a sandbox environment for educational purposes only. The site owners, publisher and the author cannot be held responsible for any damages caused.

In the post 'Target Data' means malicious actors want to steal, copy and transfer to the attacker command and control (C&C) channel or an alternative channel.

Hypertext Transfer Protocol (HTTP)

Attackers often use HTTP to exfiltrate data because this traffic is very common in enterprise networks and is always permitted. The high volume of HTTP traffic traversing enterprise networks can allow attackers to hide their evil motivation and allow data mixing with legitimate traffic.

POST is an HTTP method designed to send data to the server from an HTTP client. The HTTP POST method requests the web server to accept the data enclosed in the body of the POST message. It is often used when uploading a file or when submitting a completed web form. Usually, there is no limit to the amount of data that can be transferred using this method, except the limit imposed by the web server — if the file size is too large for the web server to handle in a single POST request, it can be split up and sent in multiple requests.

Attackers can configure the web server to respond to only specific types of requests, which allows attackers to remain stealthy. For example, the server only accepts requests from specific user-agents that are only known by the attacker.

Rising Sun is a modular backdoor malware that can send data gathered from the infected machine via an HTTP POST request to the command and control (C2) server. This malware has been observed targeting nuclear, defence, energy, and financial service companies across the world.

The following basic example of data exfiltration (Figures 1-7) relies on HTTP POST. The lab environment consists of one server as an HTTP web server with logging capabilities, and another that is considered a compromised host, which will send the stolen data using the compromised system's available tools without installing any additional software. One such tool is cURL, which is a library and command-line tool for transferring data using various protocols. When used for data exfiltration processes, cURL can POST a file to an attacker's web server from a compromised linux host, as shown in Figure 1:

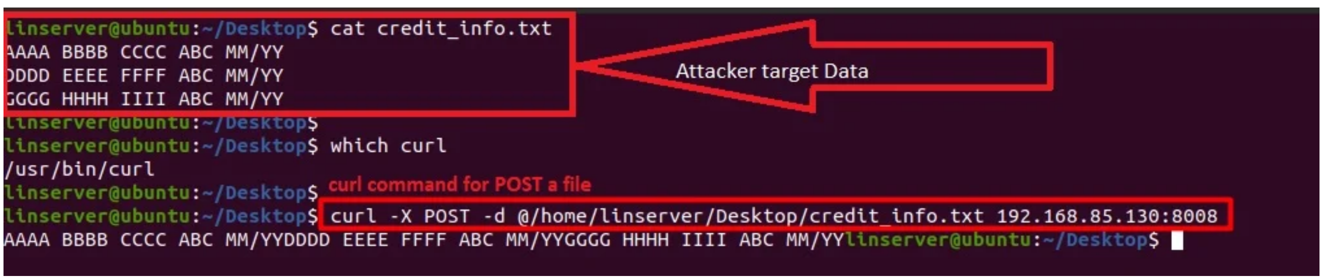

Figure 1 — cURL command for POST file to attacker server.

The attacker can then listen and capture the incoming 'Exfil Data', as shown in Figure 2:

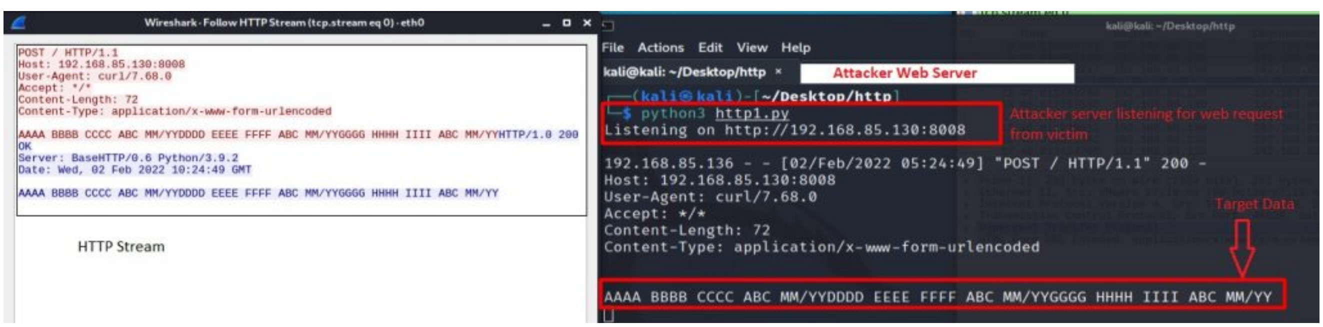

Figure 2 — Attacker server response with cURL POST command executed from the victim host.

If the victim host is a Windows platform, the attacker can use the PowerShell command to send a file using the HTTP POST method over TCP port 80, as shown in Figure 3:

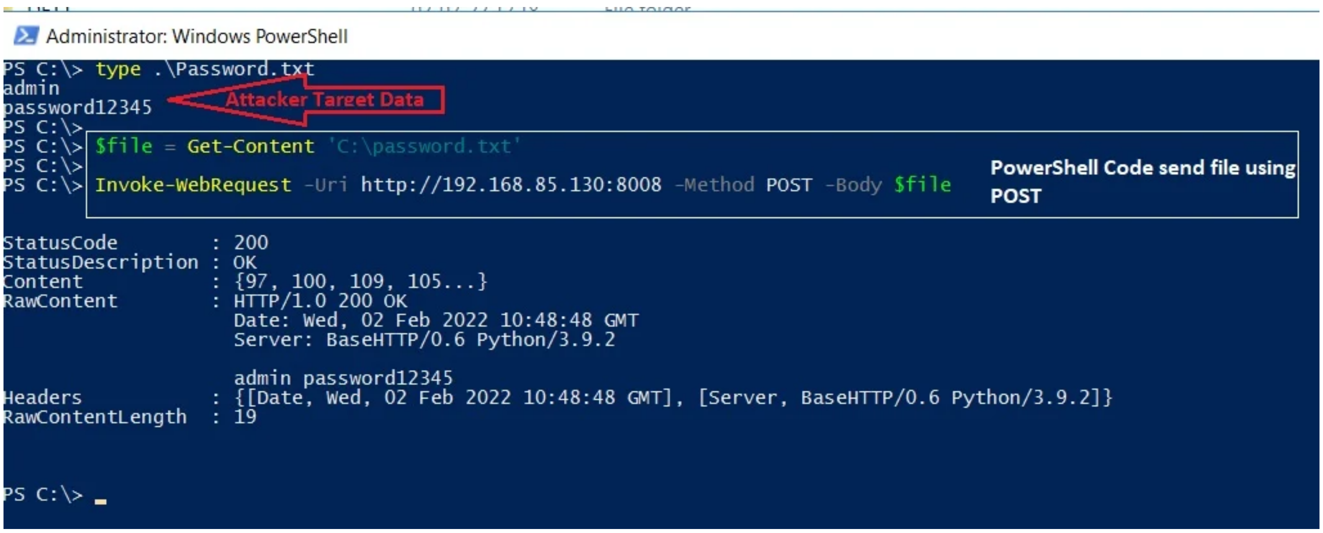

Figure 3 — PowerShell code for HTTP POST from the victim host.

The attacker server response with the above command is shown in Figure 4:

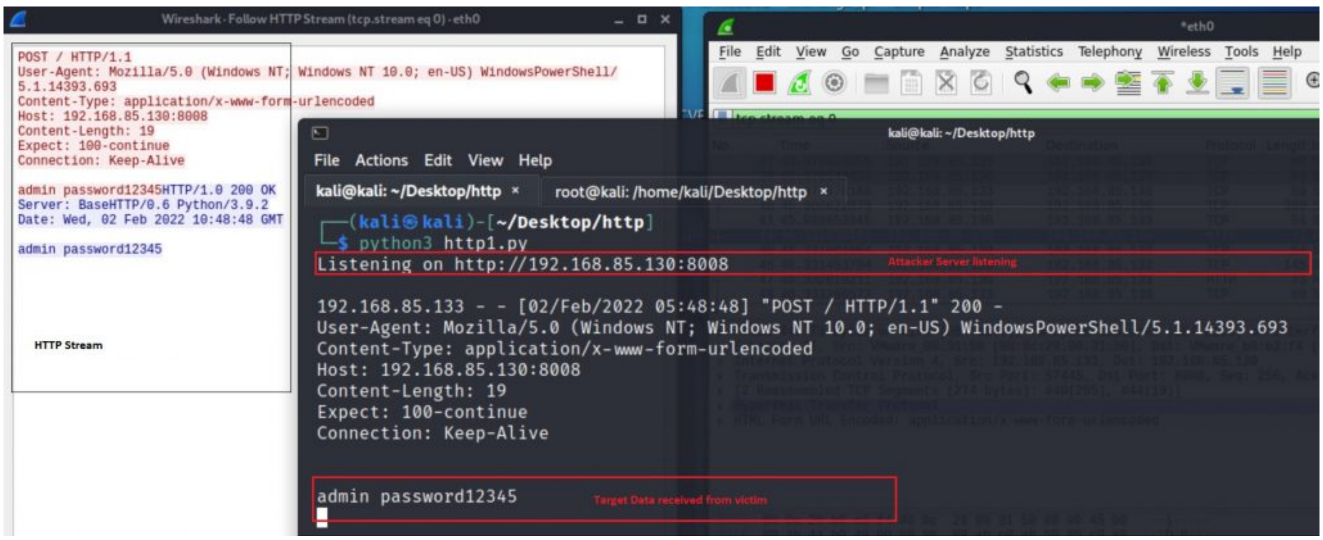

Figure 4 — HTTP POST received from the victim to the attacker listing server.

Attackers can even encrypt or compress the target data before sending it to their server (Figure 5). Most of the time this is considered as ‘stealth exfiltration’ because it seems to be legitimate traffic and usually raises no alarm to monitoring systems.

Figure 5 — Victim host to attacker control system communication with encrypted Payload using HTTP POST.

After the PowerShell code is executed, the following HTTP POST, along with encrypted payload/data requests, are sent to the attacker's server (Figure 6).

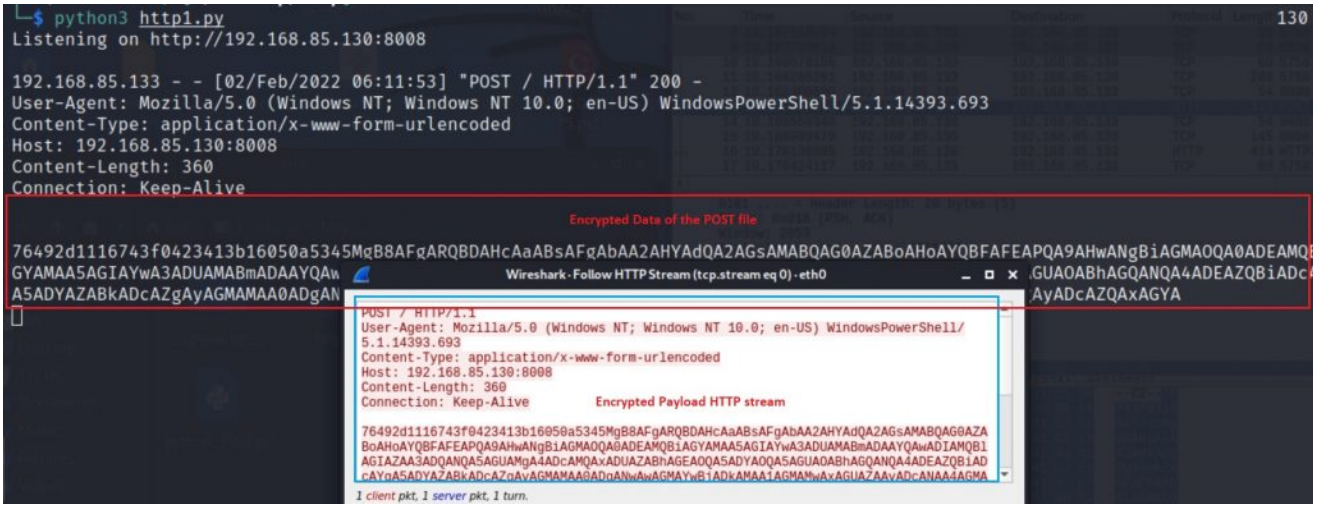

Figure 6 — HTTP POST along with encrypted payload received from the victim to the attacker listing server.

Decrypting the data is an easy job for an attacker as they know the key (Figure 7).

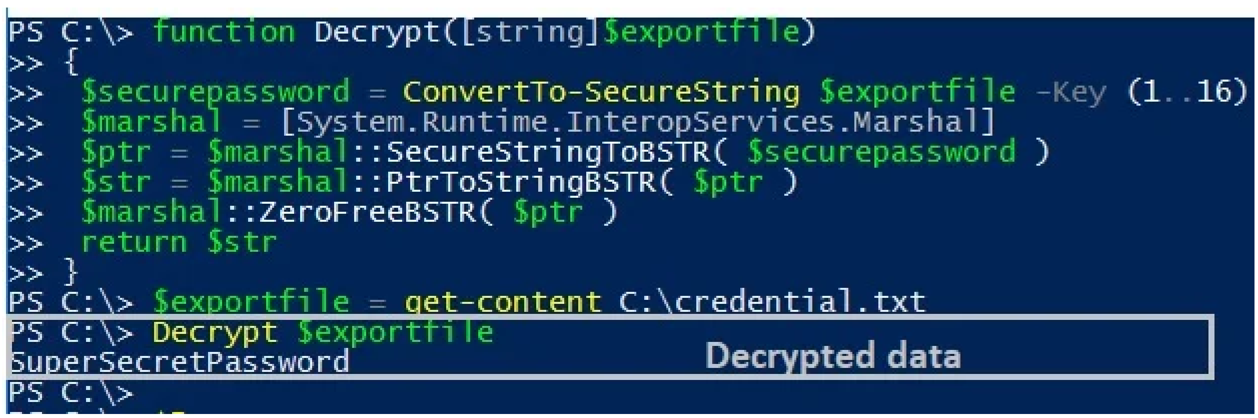

Figure 7 — Encrypted payload/data decryption.

The Simple Mail Transfer Protocol (SMTP)

SMTP is one of the most common methods for data exfiltration. Several malware programs exfiltrate the stolen information to an attacker-controlled SMTP server. For example Agent Tesla is a Windows-based keylogger and Remote Access Trojan (RAT) commonly uses SMTP to exfiltrate stolen data.

Attackers can use the following PowerShell code (Figure 8) to send an email with an attached file (stolen data) to exfiltrate a remote address:

Figure 8 — Victim host sends stolen data to the attacker-controlled email box using SMTP.

Figure 9 shows the 'Successful Email Delivered' to the attacker's email box:

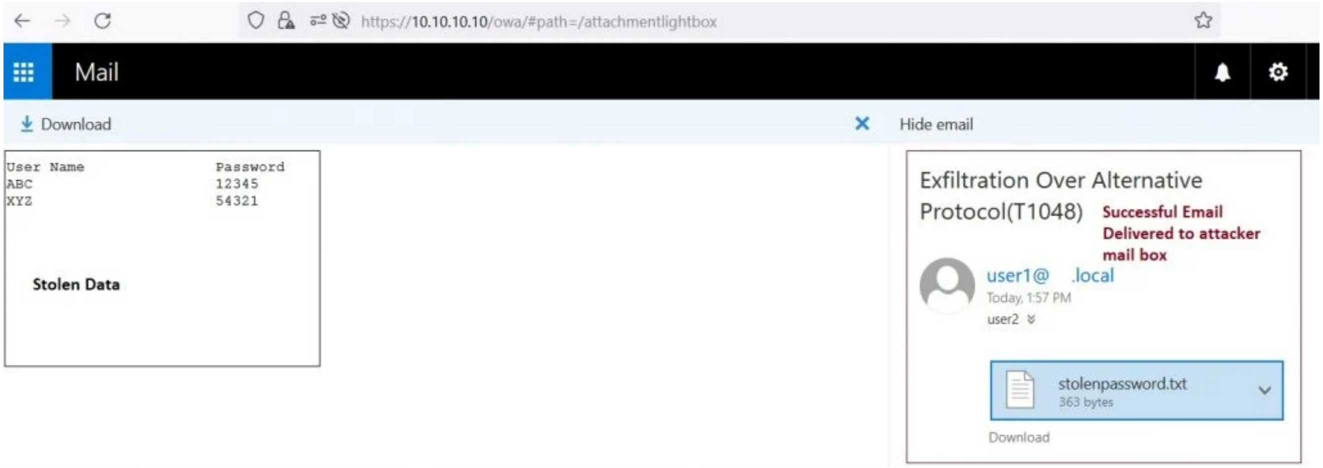

Figure 9 — Exfiltrated data delivered to attacker's email box.

The related SMTP streams are shown in Figure 10:

Figure 10 — SMTP streams.

Domain Name System (DNS)

The DNS is the hierarchical and decentralized naming system used to identify computers, services, and other resources reachable through the Internet or other Internet Protocol (IP) networks. This protocol works through TCP/UDP port 53 by default and is used only to exchange specific data.

During the exfiltration phase, the attacker makes a DNS query (initiates a domain name resolution request) to an external DNS server address. Such requests are not usually blocked by security perimeters. Without responding with an A record in response, the attacker's name server will respond with CNAME/MX or a TXT record that allows the data to be sent between them and the victim.

The attacker can then use the following tools to extract files:

DNSMessenger is a RAT used to conduct malicious PowerShell commands on compromised computers. DNSteal is a tool that creates a fake DNS server that allows attackers to stealthily extract files from a victim machine through DNS requests. This tool also supports Gzip file compression.

Figure 11 shows the attacker running the DNSteal tool with a -z parameter (transferred file was compressed/zipped by default in their attacker-controlled name server):

Figure 11 — The DNSteal tool showing the attacker's name system.
From the victim system (Figure 12), it tries to send the targetdata.txt file over the DNS connection using the following command:

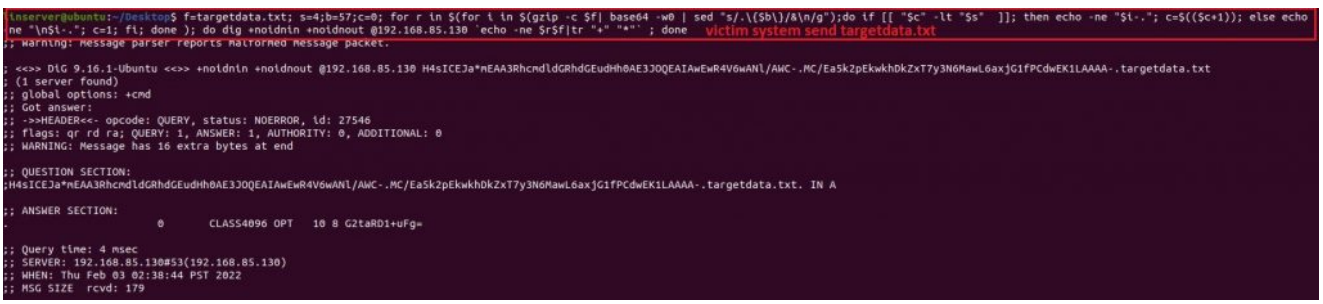

Figure 12 — The victim system sends the targetdata.txt file over the DNS.
Finally, the attacker's name server receives the target data (Figure 13).

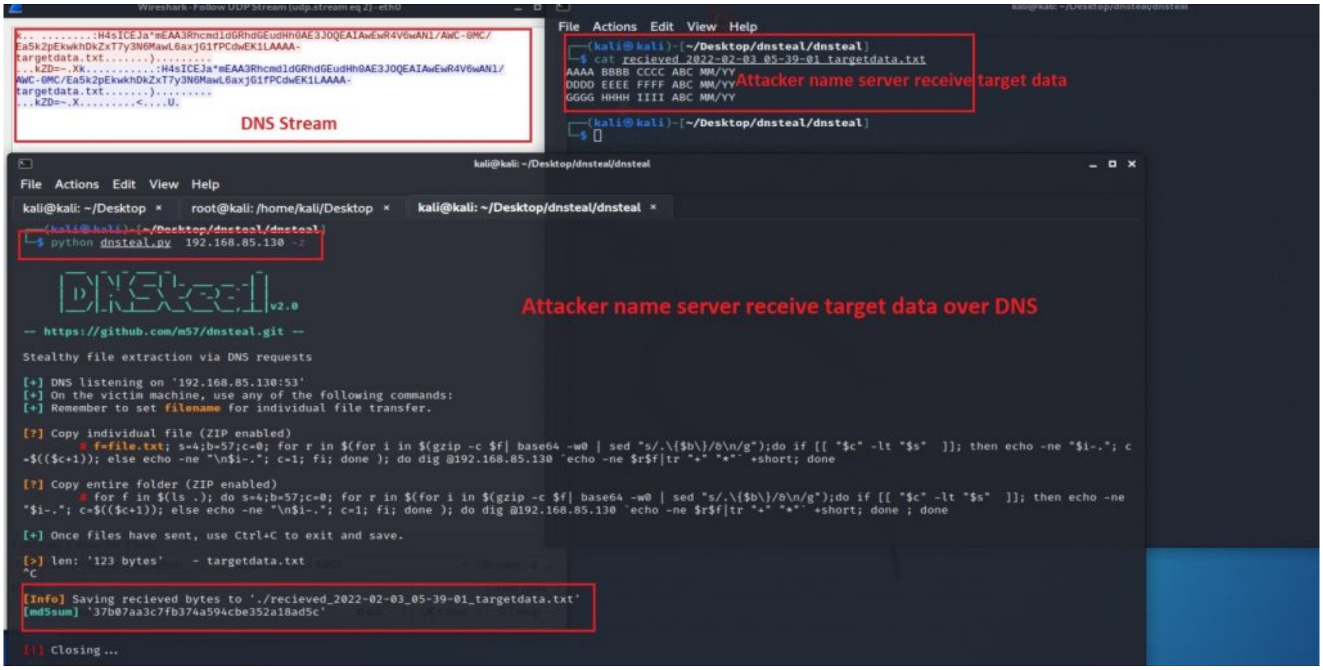

Figure 13 — The attacker’s name server receives the target data using DNS as a communication protocol.

Internet Control Message Protocol (ICMP)

ICMP is a supporting protocol in the Internet protocol suite and is widely known for its applications such as ping or traceroute. Malicious actors can use ICMP to exfiltrate data, by taking advantage of organizations that allow outbound ICMP traffic. A common example of this is the Windows malware Pingback, which uses ICMP for its C2 activities.

Metasploit have a module that will receive the exfiltrated data over ICMP from the victim host. For exfiltrated data, it needs a tool name Nping (Nping comes with Nmap). The Metasploit module server-side component receives and stores files exfiltrated over ICMP echo request packets.

Figure 14 shows the Metasploit listener machine on the attacker’s side, following the load and run module:

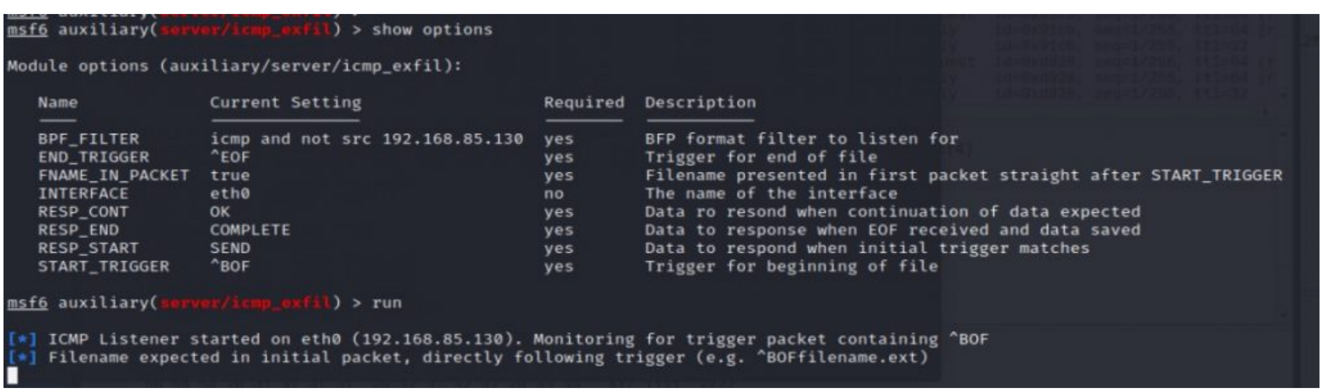

Figure 14 — Metasploit module load and run for ICMP ping replies from the victim.

Figure 15 shows an example of the victim host using the following command:

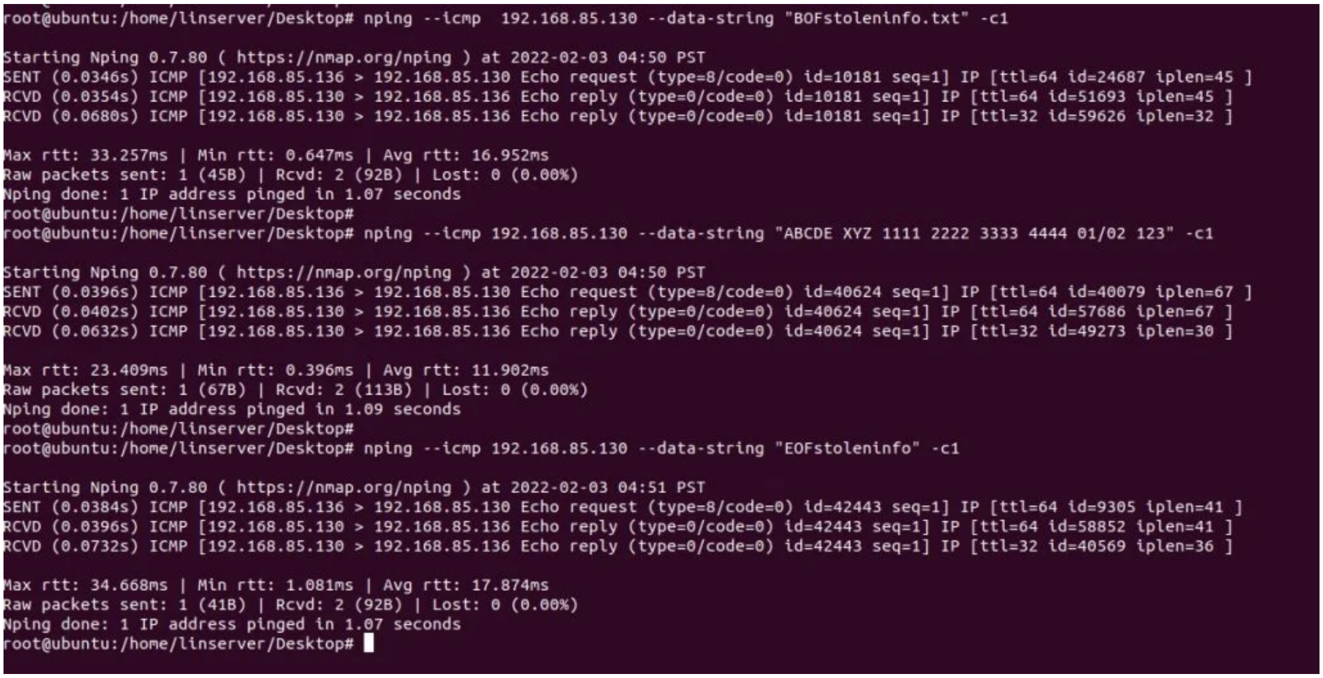

Figure 15 — Victim uses the Nping tool to send stolen data using ICMP.

In the first command, Nping will send data via ICMP. This will show up as stoleninfo.txt in the attacker's machine. The second command sends the target data and final packets containing the 'EOF' string. This tells the Metasploit module that data exfiltration over ICMP is completed.

In the attacker Metasploit module (Figure 16) we found the following:

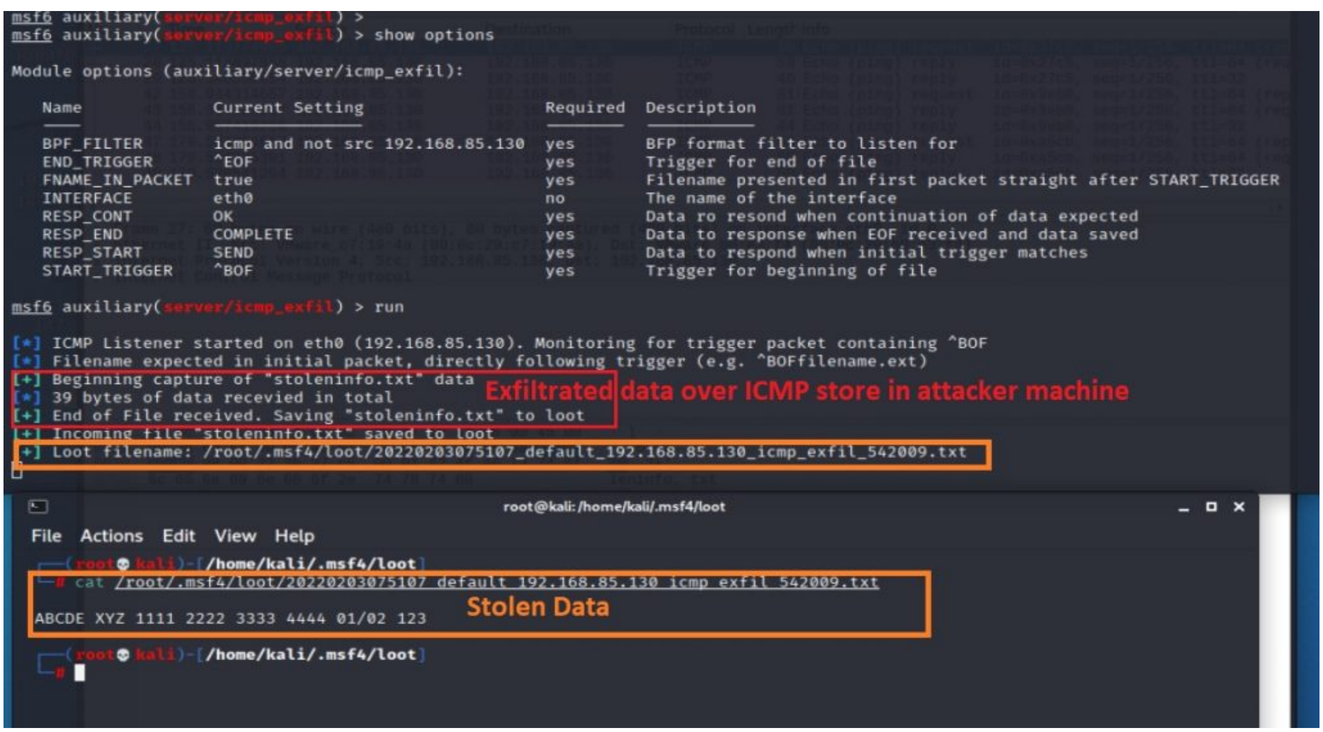

Figure 16 — Exfiltrated data stored in the attacker machine using ICMP.

The packet capture shows data is transferred via the ICMP protocol (Figure 17):

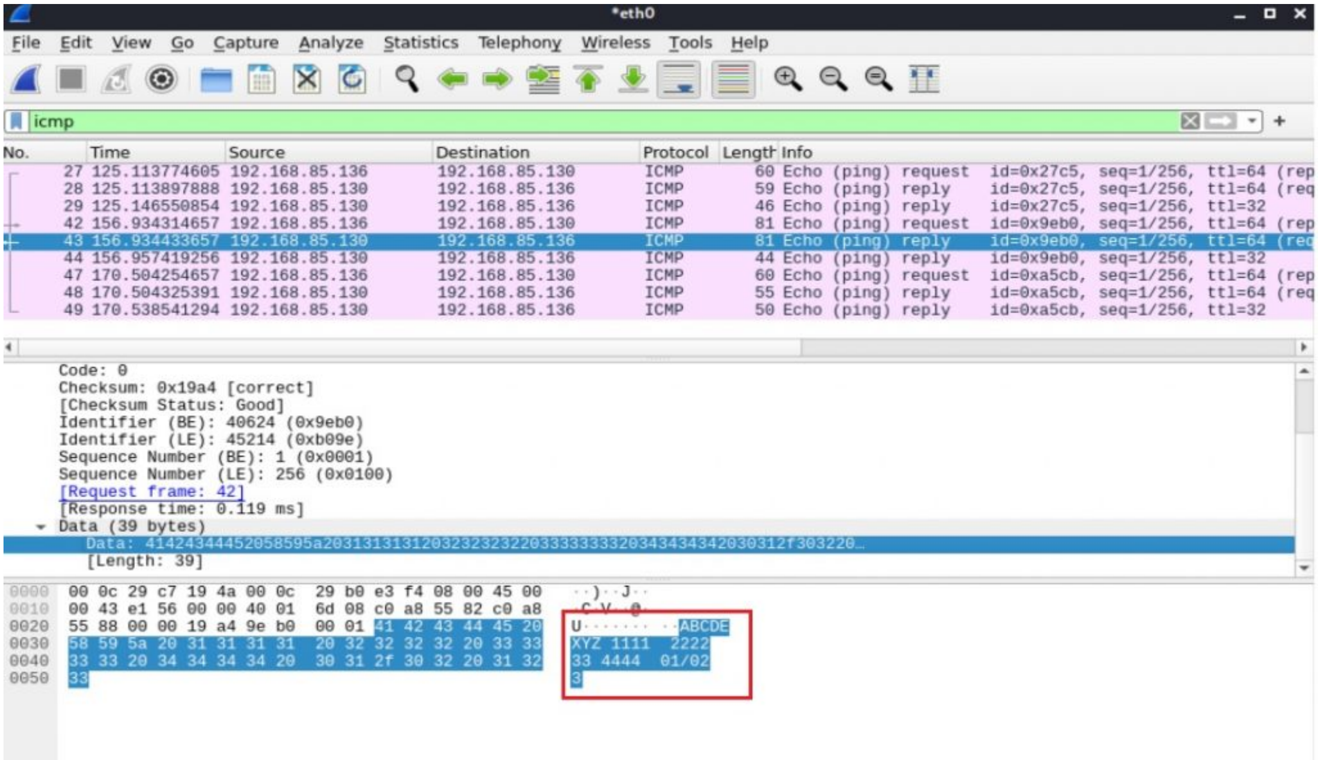

Figure 17 — ICMP data exfiltration packet capture.

Address Resolution Protocol (ARP)

ARP is a communication protocol used for discovering link-layer addresses, such as a MAC address, associated with a given Internet layer address. The ARP protocol also allows data to be transferred in local networks (outside the Local Area Network (LAN) it will not work).

An attacker can exfiltrate data via the ARP protocol using ARPExfiltrator. This tool has two parts:

1. Sender script, which runs on the victim machine.
2. Receiver script, which runs on the attacker's machine with root privilege.

The sender encodes the string buffer using the base64 algorithm and sends each letter of the encoded string as a network IPv4 address. The receiver matches each letter with a shared list of IPv4 addresses and decodes the received base64 encoded string.

The process starts with a 'receiver script' on the attacker's machine (Figure 18):

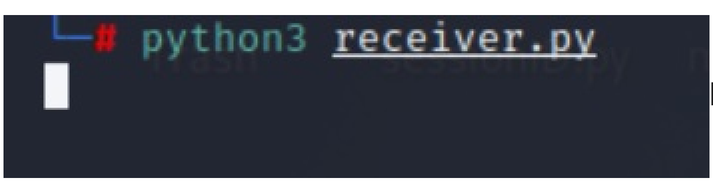

Figure 18 — 'Receiver script' running on the attacker's machine.

The victim host then tries to send the /etc/shadow file using the sender script (Figure 19):

Figure 19 — Victim sends /etc/shadow content to the attacker host using ARPExfiltrator.

The attacker server receives the stolen data from the victim using ARPExfiltrator (Figure 20):

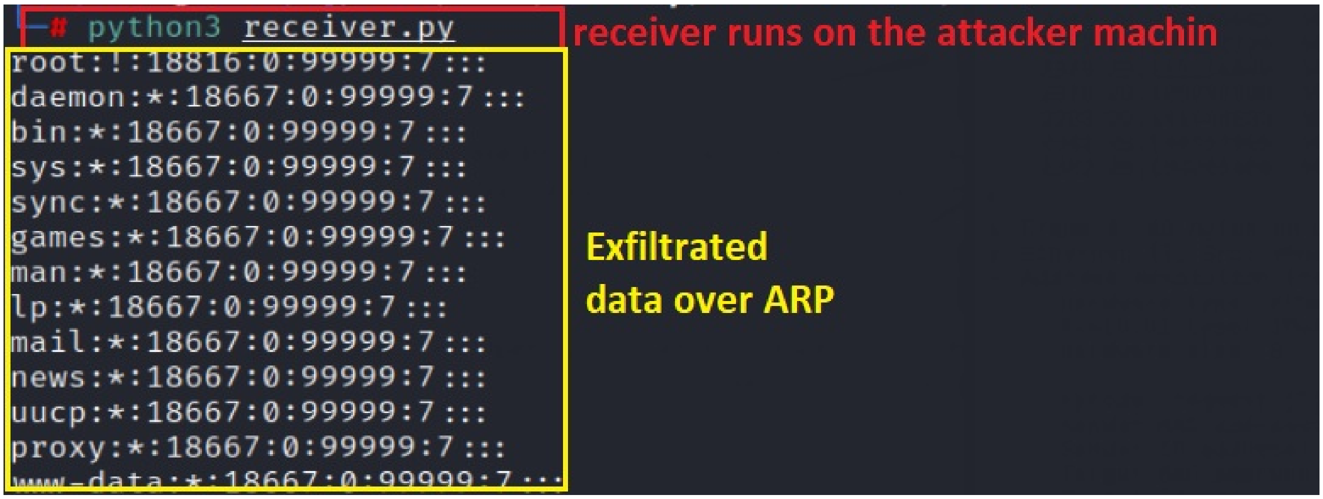

Figure 20 — Attacker server listens for exfiltrated data over ARP.

The sender script enables a 1 command in ARP Opcode — the Opcode field in the ARP message specifies the nature of the ARP message where 1 is for the ARP request and 2 is for the ARP reply. This results in a huge number of ARP request messages being sent in the LAN, as can be seen in Figure 21:

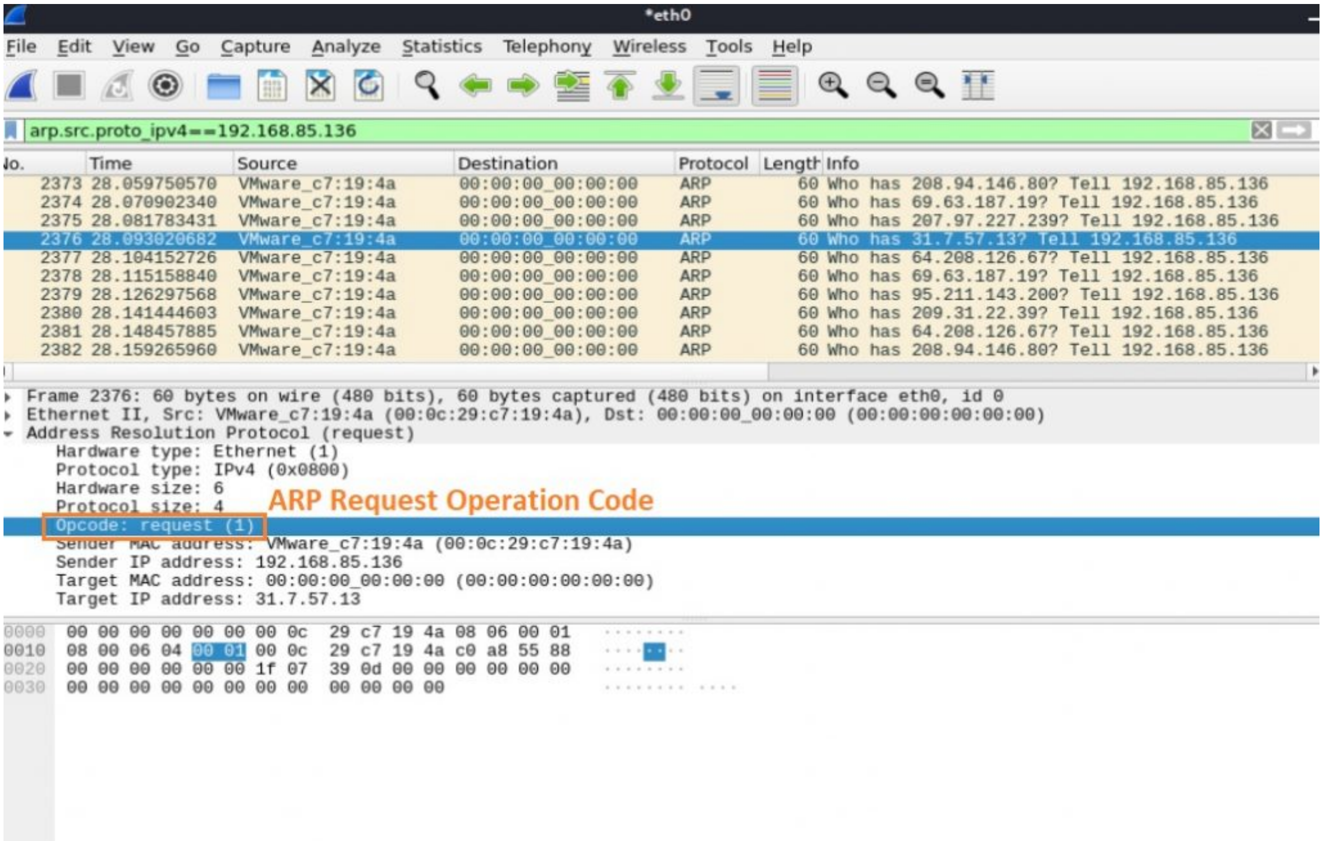

Figure 21 — ARP request operation code.

Exfiltration using IPv6

IPv6teal is a Python 3 tool that exfiltrates data from an internal network using a covert channel built on top of the IPv6 header flow label field.

A flow is a group of packets, for example, a TCP session or a media stream. The special flow label 0 means the packet does not belong to any flow. The purpose of the flow label is to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. Basically, it is designed to avoid reordering of data packets.

The IPv6teal tool can build a covert channel by storing data to exfiltrate in this field. The exfiltration script sends one IPv6 packet per 20-bits of data, and the receiver script reconstructs the data by reading this field. The payload of every IPv6 packet sent contains a magic value, along with a sequence number, so the receiving end can determine which IPv6 packets are relevant for it to decode.

Figure 22 shows the IPv6teal 'sender script' running on a victim host:

Figure 22 — IPv6teal 'sender script' running on a victim host.

Figure 23 shows the ‘IPv6teal receiver script’ running on the attacker-controlled machine; it is trying to listen and capture the targeted data:

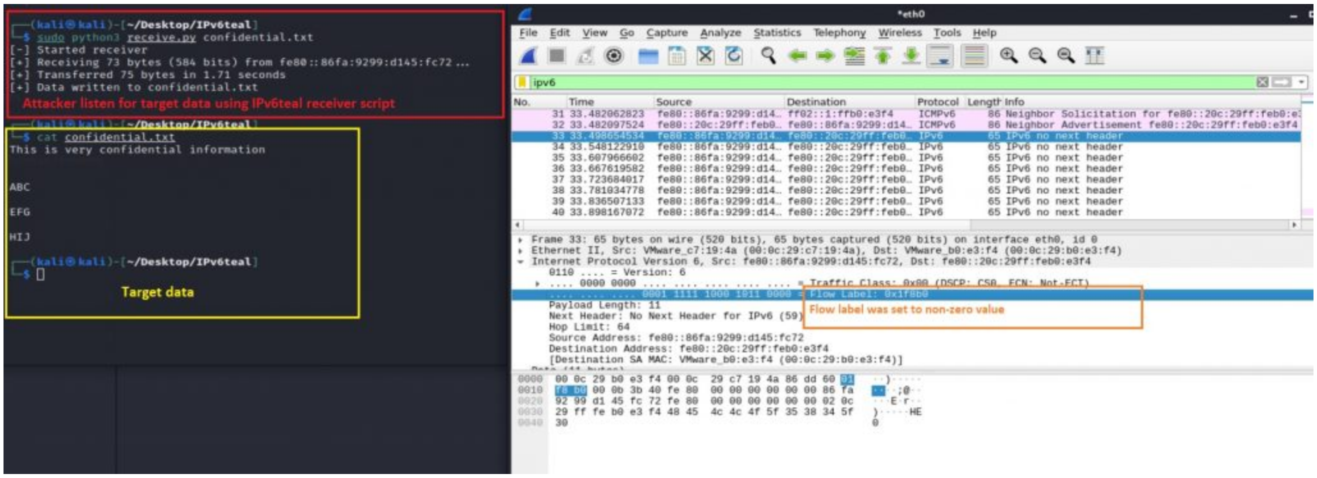

Figure 23 — ‘IPv6teal receiver script’ and related streams.

Best practices for detecting data exfiltration

Detecting data exfiltration can be a difficult task and depends largely on the type of attack method used. Cyber attackers use various sophisticated techniques, including various legitimate processes that are more difficult to detect. Consequently, analysts can mistakenly mark the data exfiltration traffic as regular network traffic.

To detect the presence of a malicious actor, more and more organizations are using automated tools that detect in real-time malicious or unusual traffic automatically. The Security Information and Event Management System (SIEM) is one such tool that can monitor network traffic in real-time. Some SIEM solutions can even detect malware being used to communicate with C2 servers.

Other best practices include:

- **Monitoring for outbound traffic patterns** as malware needs to regularly communicate with C2 servers to maintain a consistent connection. Continuous monitoring provides opportunities to detect data exfiltration with common protocols such as HTTP:80 or HTTPS:443. It’s worth keeping in mind that some advanced malware randomize delays between C2 communications.
- **Monitoring the volume and frequency of data transmission by organization users over email**. To do this we first need to calculate the average amount of data that internal users send per day. If this average data size is exceeded (say by 5 or 10 times), this triggers an alert to be investigated.
- **Keeping an up-to-date log of all approved IP addresses connections to compare against all new connections**. Along with this, it’s advised to keep an eye out for large data flows to unexpected IP addresses and major spikes in anomalous outbound traffic.

Most of these practices require searching for known attack signatures and anomalies. From this information, analysts can also build out the entire sequence of an event and map them to a known attack framework.

Best practices for preventing data exfiltration

We can divide most effective preventive measures into three categories: Preventative, Detective, and Investigative.

For example, we should ensure that only known acceptable services are permitted into the network. If suspicious network services are running then effective detective controls can trigger alerts, so analysts can investigate and take the appropriate measures immediately.

Preventative controls include: implementing and maintaining technical controls like ACL; deception techniques; encryption of data in process, transit, and at rest; host-based auditing for identified security weaknesses; and remediation.

Investigative controls include various forensics actions as well as gathering intelligence operations, so security teams can improve their knowledge base and create a custom detection system that meets organizational-unique risk profiles.

The following are a few easy methods to prevent data exfiltration from occurring in your network:

- **Employee terminations**: The Computer Emergency Response Team (CERT) at the Software Engineering Institute of Carnegie Mellon University produced a paper showing that employees were more likely to engage in data exfiltration when they anticipated imminent termination. Prohibiting an employee’s access to IT systems should happen immediately whenever an employee’s contract is over/ended/terminated. The same is true of business partners or vendors.
- **Block unauthorized communication channels**: First, disable all unauthorized communication channels, ports and protocols by default, and re-enable on an as-needed basis.
- **Create a baseline of normal data flows**: This includes amounts of data accessed or transferred, and geographical locations of access against which to compare abnormal behaviours.
- **Install proper technical controls to prevent phishing attacks:** This also requires educating users about how phishing attacks work, how to detect them, and what to do when they believe they are facing one.
- **Develop Data Loss Prevention (DLP) solutions**: DLP technology can analyse the content of all data transfers to check for sensitive information against pre-existing policies to detect suspicious activity. This combined with with logging can increase the transparency of organizational data access and movement.

- **Implement data encryption and data backup processes**: Data encryption is a security method where information is encoded and can only be accessed or decrypted by a user with the correct encryption key. Encrypted data, also known as ciphertext, appears scrambled or unreadable to a person or entity accessing it without permission. Without a key, attackers have no way of understanding and using the data. Data backups help restore lost data and resume operations while the data exfiltration attack is being investigated. Encryption provides some protection by preventing access to data by bad actors.
- **Implement proper technical control**: Restrict and monitor ingress and egress to machines in the organization using networking rules, implement Identity and Access Management (IAM), set up bastion hosts, use granular permissions, and grant access to sensitive data only to those whose job function requires it.

In summary, reviewing common data exfiltration attack techniques can help analyse possible attack surfaces and related detection capabilities. It also helps to improve the threat hunting posture for an analyst, because detecting any instances of data exfiltration as early as possible gives victims a chance to minimize the impact of a breach.

Debashis Pal is an Information Security Specialist from Bangladesh.

Rate this article
