Network Traffic Analysis: To Decrypt or Not to Decrypt

By: Kiran Dhurjaty

Introduction

Security professionals will often tell you that you cannot protect what you cannot see. In talking with these professionals, you may hear statements to the effect of:

- Encryption breaks many network security solutions.
- Encryption is doing more harm than good.
- Decrypting all traffic is the only option to maintain a secure organization.

You may wonder if any of these statements are true. More often than not, as with most things in life, the answer is "it depends." It depends on the organization's risk profile, privacy, and compliance regulations, needs, and use cases.

There is no universal answer to this question, and organizations must be wary about vendors that require decryption without considering the upfront, ongoing operational and audit implications.

In this paper, we will summarize the possibilities, advantages, and disadvantages of network traffic decryption.

Brief on the TLS Handshake

Let's start with the basics: a quick refresher on the TLS handshake. Once the initial 3-way TCP handshake is complete, the TLS negotiation begins:

1. The client sends a 'Hello' message.
2. The server sends the client its certificate and public key.
3. The client verifies the server certificate with a Trusted Root Certification Authority.
4. The client and server choose the strongest possible encryption that both can support.
5. The client encrypts a Pre Master-Key with the server's public key and sends it back to the server.
6. The server decrypts the client communication with its private key, and thus is able to access the Pre Master-Key.
7. Both the client and server compute the session key from Pre Master-Key.
8. The session key (symmetric encryption) is now used to encrypt and decrypt data transmitted between the client and server.

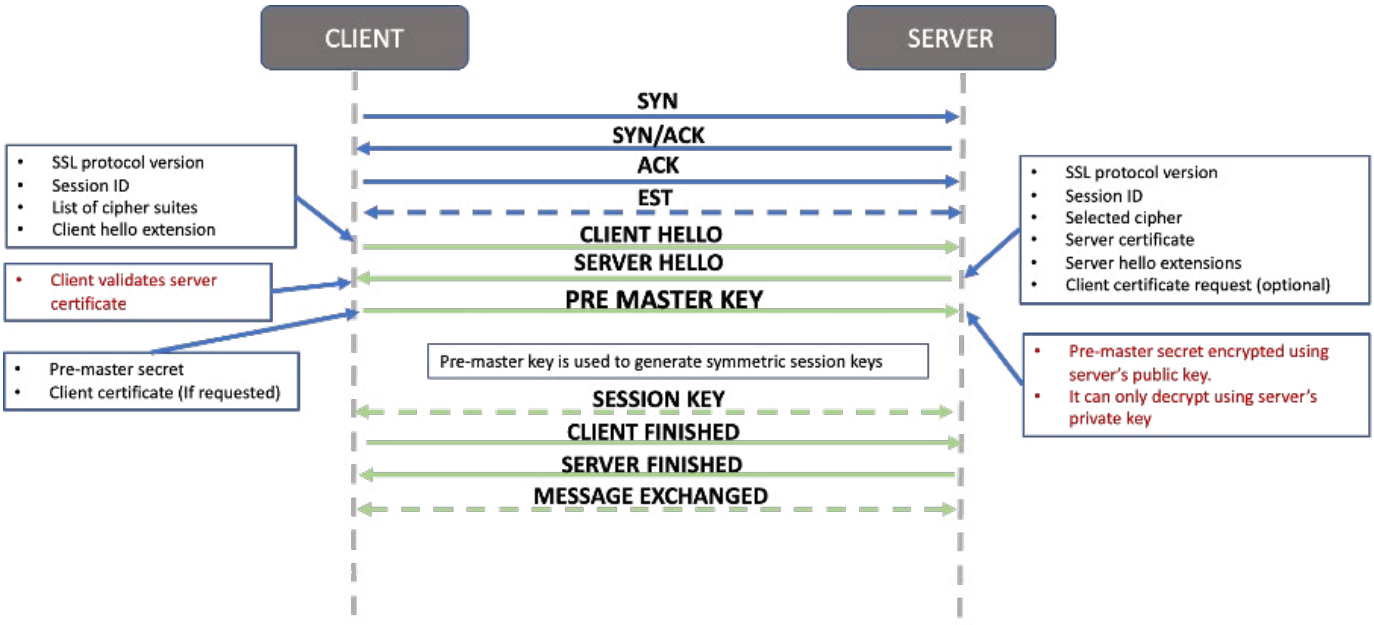

Figure 1: TLS Handshake

For the rest of our discussion, we will focus on step 3 where the client verifies the server certificate.

TLS Offloading

Let us consider a standard data center scenario. The web server holds the private key to decrypt the ciphertext based on its own public key. This implies that anyone who has this private key can decrypt the pre-master key and obtain the session key.

The typical approach therefore for TLS interception is to place an offloading engine within the data center, typically in front of the web server. This engine is given the private key which in turn allows it to decrypt the traffic and send cleartext payloads to the web server as well as any other traffic inspection tools.

These offloading engines are purpose-built devices to encrypt and decrypt at wire speed and are often used to decrease the load of these activities on the web server. Most often, the application load balancer takes up this responsibility. In this scenario, the offloading appliance must be inline with the traffic. The client establishes a connection with the offloading appliance, but the client is unaware of it as it gets a valid certificate assigned to the web server. As shown in Figure 2, the traffic from the offloading device to the web server is plaintext traffic and therefore the organization can TAP the network here to perform any traffic analysis.

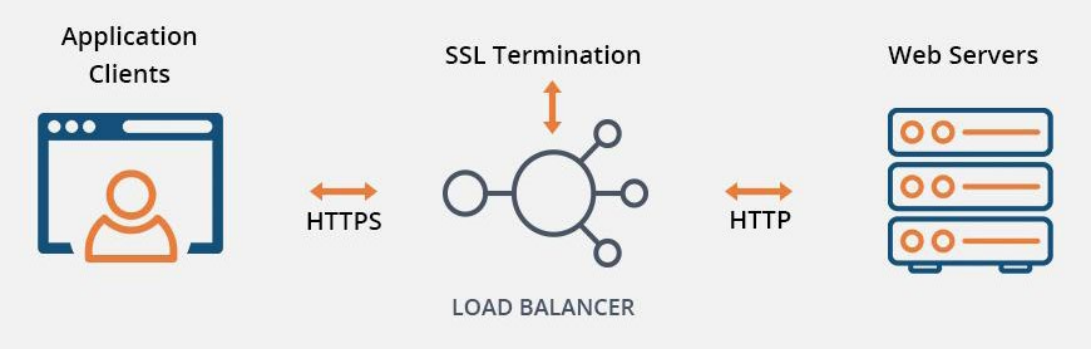

Figure 2[1]: TLS termination using a load balancer.

The other possibility is to deploy the offloading device in passive mode. This approach involves tapping the traffic before it reaches the web server and decrypting it. In this case, both the web server and passive device have the private key and are therefore able to decrypt the traffic in parallel. This passive approach is not as common since it involves sharing the private key in multiple locations and does not help offload the encryption / decryption compute load from the web server itself.

HTTPS Proxy Solution

Next, let us look at the challenges of encrypted traffic monitoring from the client side. After the TLS handshake, the client and server use symmetric-encrypted sessions for secure transport. As a consequence, no one is able to use TLS offloading outside of the organization managing the web server, since no one has the web server's private key. Therefore, if the client organization wants to decrypt the traffic, we need another mechanism.

A common possible solution is to use TLS proxy servers. These are similar to normal web proxy servers with the added feature of inspecting TLS traffic. In this case, when client browsers try to contact servers on HTTP(S) ports 80 and 443, they first hit the proxy server. For all TLS requests, the proxy server responds to the client with its own certificate and completes negotiation. It simultaneously establishes a new connection with the server that the client is looking to connect to. Thus, the proxy holds two session keys for every TLS session: the first is to decrypt traffic from the client and the second is to re-encrypt the traffic to the web server, as shown in Figure 3.

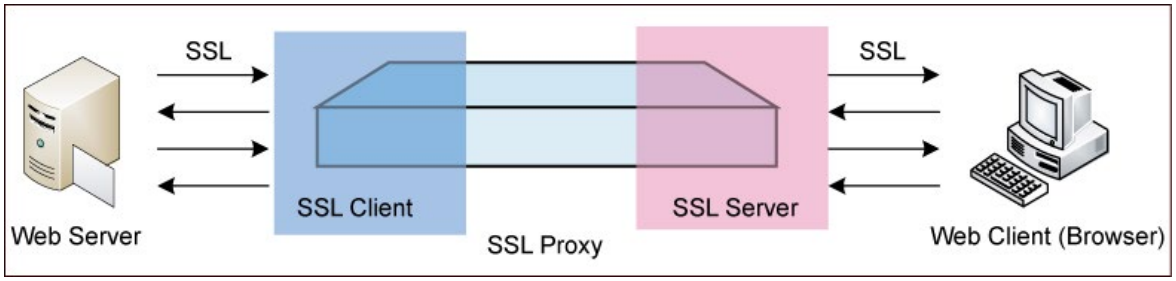

Figure 3: TLS Proxy Inspection[2]

[1] https://www.ssl2buy.com/wiki/ssl-offloading

[2] https://www.hillstonenet.com/support/4.5/en/config_nbctask_sslproxy_intro.html

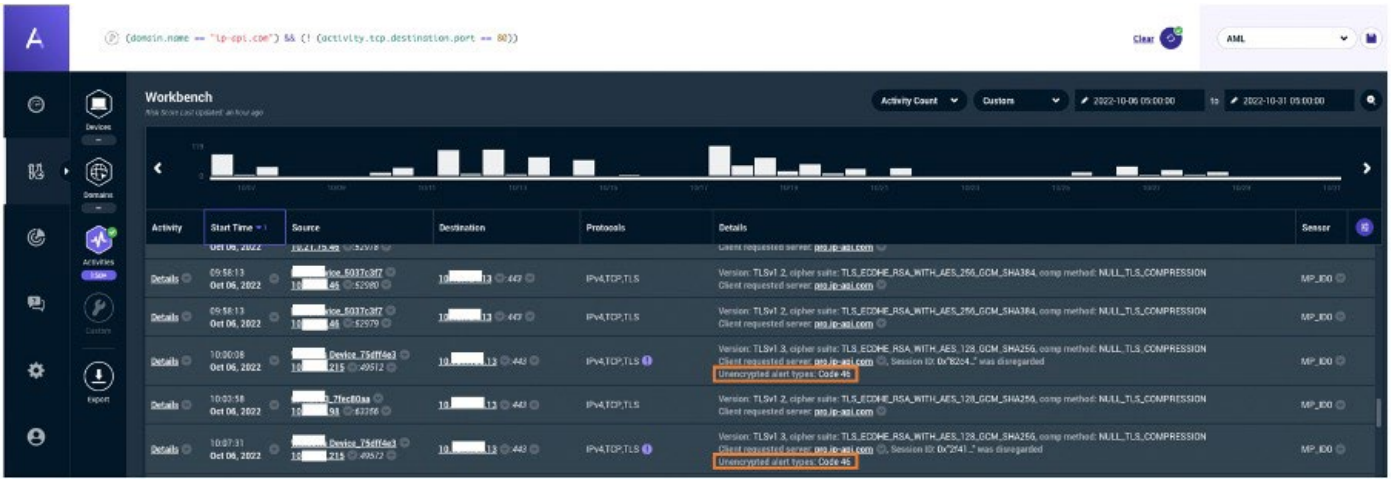

Figure 4: Certificate validation failures when using a proxy

To overcome this certificate verification challenge, organizations often create a local CA to issue trusted certificates to the proxy and then install the local CA certificate in the client endpoints' trusted certificate store.

SSL Visibility Solution

Another option is to use an SSL (or TLS) visibility appliance, which is an inline device that sees all the traffic. It is operationally similar to an HTTPS proxy (Figure 5), but with visibility to a broader set of network protocols. These devices also often use specialized hardware (ASICs) to deliver wire speed SSL visibility.

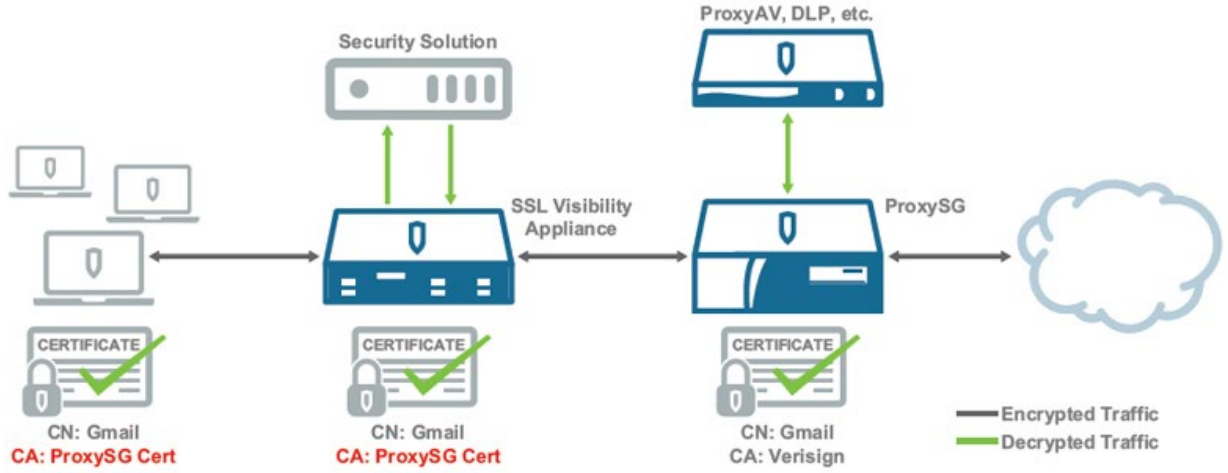

Figure 5: SSL Visibility Appliances[3]

Clearly many of these solutions empower the security team with good visibility and the ability to analyze the decrypted traffic using tools such as intrusion detection systems, data loss prevention tools and application firewalls. It also eliminates the risk of human errors in accepting invalid certificates.

[3] https://www.edgeblue.com/SV2800.asp

Should You Decrypt?

Given the background into TLS and the various decryption options, the next two items to consider are whether or not you should decrypt network traffic; and whether the benefits of the decryption justify the costs. Here are multiple considerations to help your organization decide:

- Are there privacy laws applicable to your organization either locally or because of your global presence?
- How will you make your users aware that their traffic may be decrypted? Do your corporate policies make it clear you have the right to inspect any / all traffic?
- Does the appliance / technology allow you to exclude certain traffic from inspection?
- Does your organization's data protection framework cover the handling of plain text traffic resulting from decryption without introducing new security risks?

These are just some of the legal and regulatory considerations. Organizations looking at decryption options must also consider technical and business challenges. Let's start with the total cost of ownership (TCO) of a decryption solution. When it comes to the upfront capital expenditures, you must consider the number and size of interception devices you'll need in order to decrypt the volume of traffic that you want to inspect. Alternatively, you may consider decryption as a built-in feature in some of your security tools, such as network detection and response (NDR), intrusion detection systems (IDS), next-generation firewalls (NGFW), and web application firewalls (WAF).

We see this design is especially common when an organization's security program develops in a piecemeal fashion or where each solution is run by a different team across risk, compliance, and information security. One advantage of this design is that decrypted traffic never leaves the device in question and may not need to be stored either. The disadvantage is that the keys are spread across multiple locations on your network, which in turn brings up operational security concerns.

Moreover, each of these in-line decryption solutions also add complexity, latency, and potential points of failure in the network path. This may impact not only user experience but also security and network availability. Maintaining a consistent policy across all decryption solutions is challenging since each has their own mechanism for performing functions like bypass. For example, your organization may need to avoid decryption of specific data types, such as medical records or credit card numbers. Failure to do this might add audit and compliance risk, especially with regulations/standards such as HIPAA and PCI-DSS.

Costs are also a factor of sizing: decrypting and re-encrypting the traffic involves significant computational resources. As a consequence, either traffic will only be sampled (providing only partial visibility), or these appliances will need to be double the size of what would be needed solely based on the traffic volume being monitored, which may then require a larger number of hardware components. Many vendors simply white label some of the more popular interception solutions. Ultimately, all of these factors can increase costs considerably.

Given the above challenges, most customers adopt a single TLS visibility device where you decrypt once and mirror to all other security solutions that need the unencrypted traffic.

Does Decryption Deliver Full Visibility?

Unfortunately, the answer to that question is No, because there is no universal standard for encryption. Moreover, clearly attackers are unlikely to follow recommendations and standards - they may use any cipher, including old and deprecated ciphers, or even custom encryption algorithms rather than TLS/SSL.

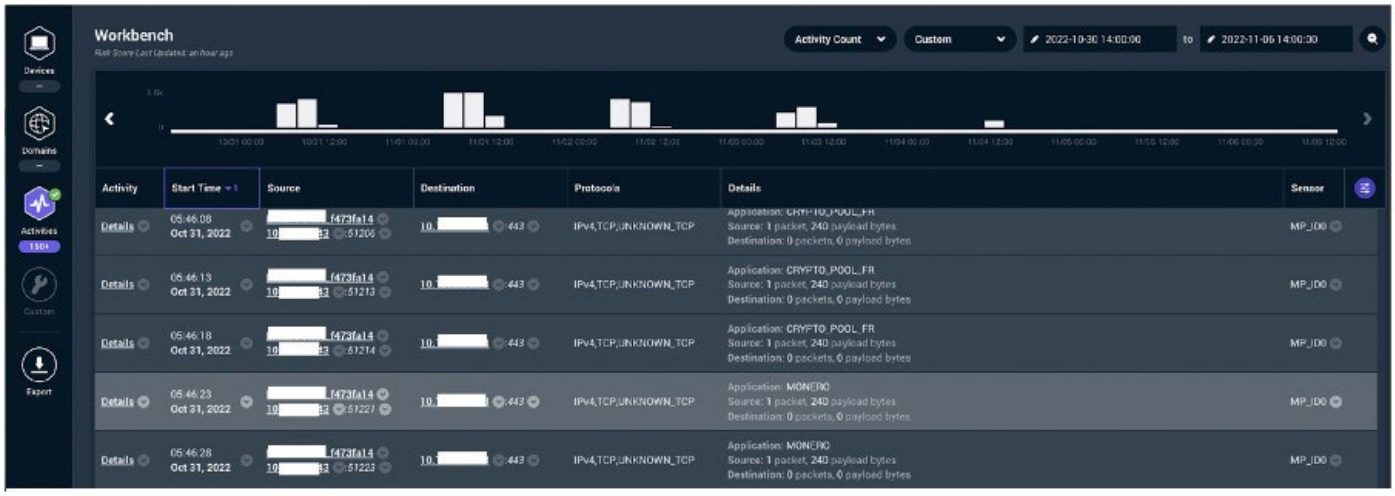

Figure 6: Cryptomining traffic

For instance, Figure 6 shows the communication from a compromised endpoint to a Cryptomining pool. While this is over port 443 (HTTPS), it is not valid TLS traffic and thus a TLS proxy is unable to identify the protocol, let alone decrypt it.

Similarly, our analysis of the command and control (C2) communications in ransomware attacks shows that attackers use both symmetric and asymmetric key cryptography for C2 communication, including hard-coded keys for initial communication so that the C2 server can share the data encryption key via a secure channel with the compromised endpoint(s). For more details on C2 communication refer to the MITRE ATT&CK framework tactic. (https://attack.mitre.org/tactics/TA0011/)

Decryption technologies also struggle with applications that use methods like certificate pinning or hardcoded server certificates. For instance, the popular messaging application WhatsApp provides end-to-end encryption for each chat but without any certificate exchange (Figure 7). Similarly, digitally signed emails cannot be decrypted by TLS interception solutions. As a consequence, it is hard to implement a policy that simply drops encrypted traffic that is not using TLS because doing this might drop a lot of traffic that should be inspected.

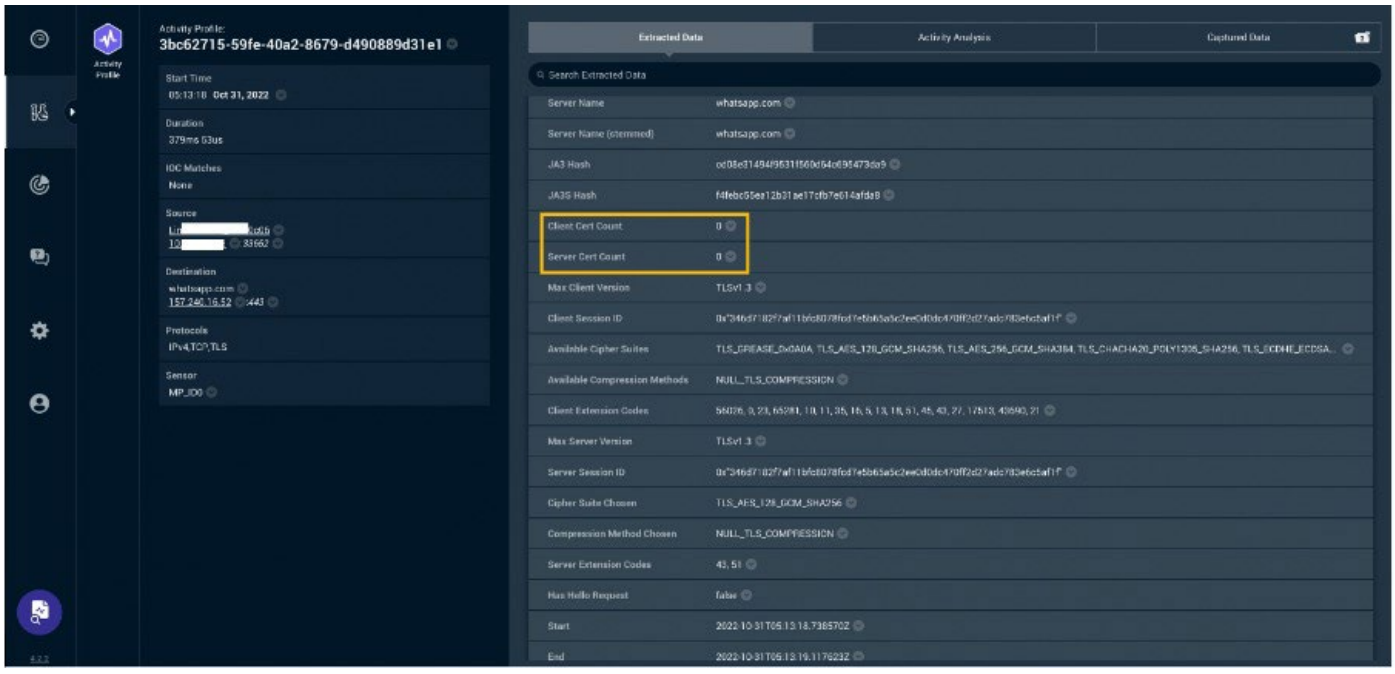

Figure 7: End to end encrypted WhatsApp traffic with no certificate exchange

Other Risks of TLS Interception

Handling sensitive data in plain text can often be more dangerous than lack of visibility. For instance, how do you prove that the decrypted data is not stored somewhere, or that access to decrypted data is sufficiently locked down? Solutions like NGFW and IDS will typically not store sensitive data, but what about the logs and alarms generated by these devices? SIEM solutions analyze and correlate logs and could therefore reveal this information. Similarly, NDR solutions are designed to unearth low and slow attacks, which need storage of metadata and packets for extended periods of time. Of course, once this data is stored it also opens the organization to insider attacks or targeted attacks that impersonate an insider with the right level of access.

How Does Arista NDR Approach Encrypted Traffic Analysis?

Arista's approach focuses heavily on the threat landscape and attacker tactics and techniques. For instance, if you look at the list of MITRE ATT&CK framework TTPs under the data source "Network Traffic"[4], you will notice that the percentage of TTPs that might benefit from decryption is a small fraction of the overall list. That coupled with the policy implications of decryption has helped Arista formulate a strategy that doesn't force the customer to decrypt network traffic.

Arista NDR uses data science to perform Encrypted Traffic Analysis (ETA) without decryption, thus working within privacy and policy constraints while also ensuring security teams do not have to fly blind. The platform uses a variety of data science techniques for this purpose, including unsupervised machine learning (ML) to identify TLS sessions with anomalous characteristics that could be C2, and supervised ML to identify patterns of activity that relate to attacker TTPs such as remote access tools, reverse shells, unauthorized command and control, domains used for data exfiltration, etc. Arista NDR also uses deep neural networks and decision trees for classifying encrypted sessions based on the nature of traffic such as remote shell activity, web browsing, video conferencing, file transfers, etc., as shown in Figure 8.

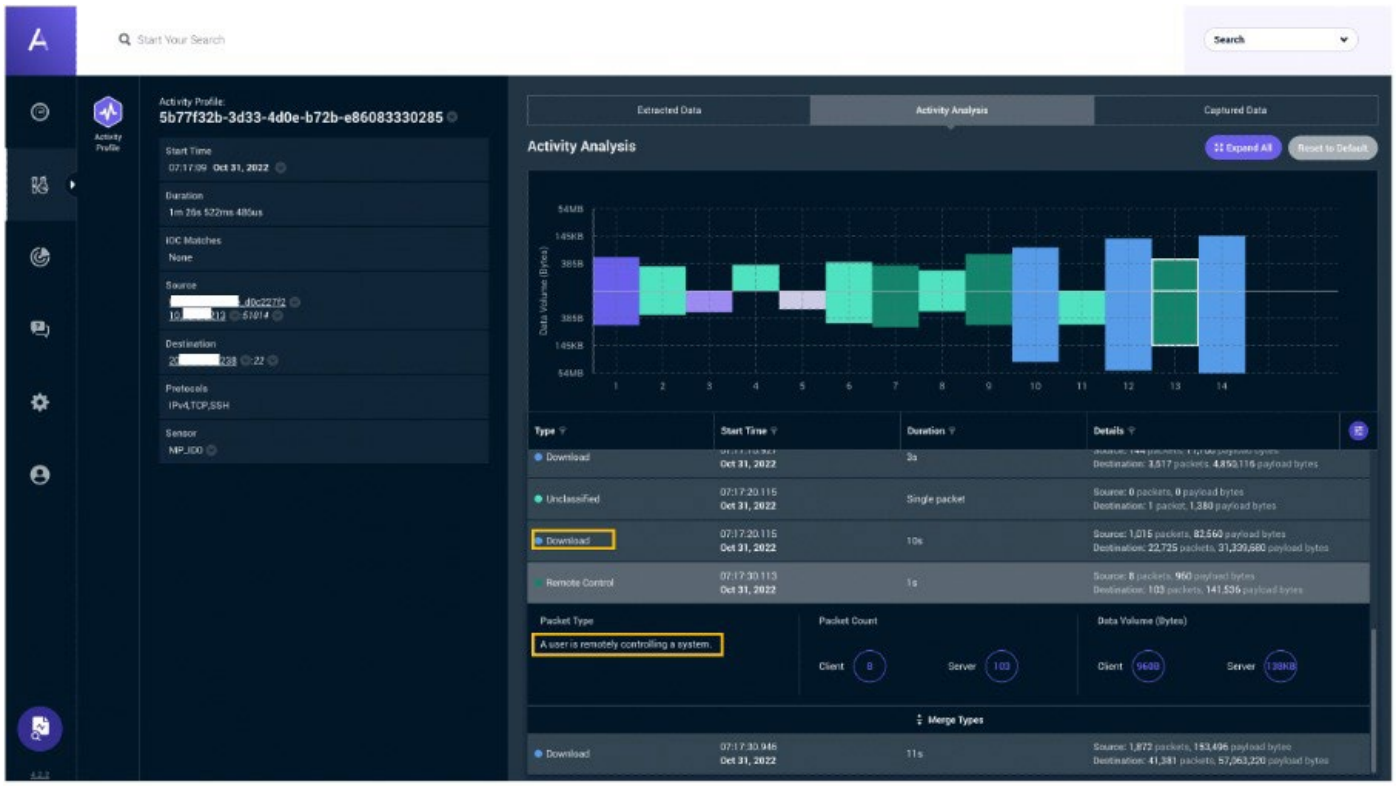

Figure 8: SSH Activity Analysis

[4] https://attack.mitre.org/versions/v12/datasources/DS0029/

In conjunction with these methods to analyze encrypted traffic, other context such as the nature of the source application/device, destination domain, IP address, autonomous system networks, and other network parameters are highly relevant and helpful to the analyst. For example, it might not be a concern when an encrypted file is transferred via email or over a Zoom meeting session between internal users. On the other hand, if that file is transferred from a PowerShell script or an IoT device to a Dropbox account rather than your corporate file-sharing application, it would be far more concerning.

Other security products utilize techniques such as JA3, JA3S, or JARM hashes, in combination with indicators of compromise and other forms of threat intelligence, to identify threats without the need for decrypting the traffic. This by itself can be a noisy detection technique. However, when combined with Arista NDR's more thorough ETA engine, the results have been impressive given they come with the advantage of avoiding the decryption baggage. In fact, we would argue that some detections are possible only when analyzing the encryption process and protocol information! For instance, Arista NDR analyzes the usage of invalid TLS certificates or those issued by free certification providers, since this is often a tell-tale tactic used by attackers.

Summary

In conclusion, organizations must exercise caution before deciding to decrypt network traffic. Consider the tradeoffs ranging from cost, privacy, ongoing operational challenges, audit, and insider threats, to name a few. In evaluating their risk profile, most organizations find decryption is not the best approach since techniques like encrypted traffic analysis deliver a greater benefit at a lower cost and with less effort while avoiding the risks and difficulties introduced by decryption.

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office
1390 Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062