Using AI to Detect Malicious C2 Traffic

By Ajaya Neupane, Stefan Achleitner

Published: 2021-05-24 · Archived: 2026-04-05 22:01:43 UTC

Executive Summary

Sophisticated malware, such as Emotet and Sality, and advanced persistent threats (APTs), such as the recent SolarStorm attack, emphasize the necessity for advanced detection methods to identify novel, unknown types of malicious network traffic.

Current intrusion prevention systems (IPS) typically work based on signature matching and monitoring network traffic for known patterns in the data packets. Such static methods fall short in detecting unknown types of malware-generated network traffic, which calls for more advanced detection techniques that incorporate inspection of the overall packet structure, rather than specific static patterns.

In a blog on data leakage from Android apps, Unit 42 researchers demonstrated that unknown traffic types that leak sensitive user information could be detected using machine learning techniques.

Based on command and control (C2) traffic from malware, such as Sality and Emotet, this blog analyzes how deep learning models are further able to identify modified and incomplete C2 traffic packets. This analysis illustrates that the usage of machine learning techniques in IPS can discover yet unseen variants of C2 traffic and can help detect advanced attack campaigns.

Palo Alto Networks Next-Generation Firewall customers are protected from such types of attacks by IPS and AppID in our Threat Prevention security subscription and with malware analysis and prevention through our WildFire security subscription.

C2 Attacks

One of the most damaging aspects of malicious network attacks is accomplished through C2. After malware infects a computer, it establishes a connection to the attacker's server -- the so-called C2 server -- to perform additional tasks that may include downloading other malicious software, data theft or establishing remote control.

In the following sections, we introduce several malicious C2 traffic types, which we use as samples to show how an advanced machine learning system can detect such traffic. The discussed malware serves as examples to illustrate the effectiveness of our machine learning AI in the detection of C2 traffic. The detection capabilities of our AI are not limited to the presented malware samples, but can be applied to general C2 detection.

Sality

The Sality malware was first discovered in 2003 and became more advanced over the years due to the continuous development of new features and capabilities. Sality spreads itself by infecting and modifying executable files and

copying itself to removable drives and shared folders.

Once the malware infects a computer system, it attempts to open connections to remote sites, download additional malicious files and leak data from the host machine. Although Sality has been around for a while, the continued development and addition of new features make it an effective and complex malware.

The following two HTTP packet headers (Figures 1 and 2) show C2 traffic used by Sality to connect to the remote site *padrup[.]com.*

```
GET /sobaka1.gif?12db3cf=98861835 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0b; Windows NT 6.0)
Host: padrup[.]com
Cache-Control: no-cache
Cookie: jsessionid=85b50d8fab658ecb9f79aa4de6039c87
```

Figure 1. Sality C2 traffic.

```
GET /sobaka.aspx?24c1882=115624326 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0b; Windows NT 6.0)
Host: padrup[.]com
Cache-Control: no-cache
Cookie: jsessionid=a2b0f43b9876d289325c3f13a7f8f95b
```

Figure 2. Sality C2 traffic.

C2 traffic from Sality, such as the packets shown in Figures 1 and 2, communicates with various C2 servers worldwide to perform tasks such as downloading and installing additional malware or leaking sensitive data.

Emotet

Emotet malware has been known since 2014 as banking malware. Typically, Emotet is distributed with Microsoft Word documents containing embedded macros to infect vulnerable hosts. C2 traffic from Emotet malware transmits encoded or otherwise encrypted data over the HTTP protocol. In Figures 3 and 4, we show HTTP packet headers from Emotet C2 traffic.

```
POST /r1s4dvgwanu1ov8qku/e6qj08nos8kh/o7rhpr2xi05tkkp/ HTTP/1.1
DNT: 0
Referer: 90.[]160[.]138[.]175/r1s4dvgwanu1ov8qku/e6qj08nos8kh/o7rhpr2xi05tkkp/
Content-Type: multipart/form-data; boundary=----------------------1BetPUScZnIzXogZ6qQcQ8
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3[.]0[.]30729; Media Center PC 6.0; .NET CLR 1.1.4322;
.NET4.0C; .NET4.0E; InfoPath.3)
Host: 90[.]160[.]138[.]175
Content-Length: 5556
Connection: Keep-Alive
Cache-Control: no-cache
```

Figure 3. Emotet C2 traffic.

POST /kl4or/ok48hg/a5msy52s4i4uuac7dm/pzudacb2/a51azs1nbhzmu5m/p0f6wimb1tcqvn0/ HTTP/1.1
DNT: 0
Referer: 184[.]66[.]18[.]83/kl4or/ok48hg/a5msy52s4i4uuac7dm/pzudacb2/a51azs1nbhzmu5m/p0f6wimb1tcqvn0/
Content-Type: multipart/form-data; boundary=---------O8dHD39IM
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET CLR 1.1.4322; .NET4.0C; .NET4.0E; InfoPath.3)
Host: 184[.]66[.]18[.]83
Content-Length: 6916
Connection: Keep-Alive
Cache-Control: no-cache

Figure 4. Emotet C2 traffic.

Further details about Emotet's C2 traffic and how to analyze it can be found in Unit 42's posts, Wireshark Tutorial: Examining Emotet Infection Traffic and Attack Chain Overview: Emotet in December 2020 and January 2021.

Detecting C2 Traffic

The goal of an IPS is to accurately identify connections to a C2 server. Due to the dynamic nature of the internet and the fast-changing assignment of IP addresses and domain names, this is very challenging to achieve, and defenders often lag behind attackers.

An approach typically used in today's security industry is to identify C2 traffic, such as network packets from Emotet, as shown above, with static signatures that match a specific pattern in the traffic. This approach has the advantage of being accurate, but it is not flexible in detecting variations or unknown types of traffic. Detecting packets (shown in Figure 3) are especially problematic since there are no reliable patterns in the packet that could be used in a signature. For example, the Uniform Resource Identifier (URI) path from an Emotet packet (shown below) appears to contain random strings, which might transmit encoded information, but would not be a reliable pattern to be used in a signature:

POST /r1s4dvgwanu1ov8qku/e6qj08nos8kh/o7rhpr2xi05tkkp/ HTTP/1.1

A similar case can be observed in the user-agent field, which shows generic values from a web browser, as well as the hostname, which consists of a specific IP address:

User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET CLR 1.1.4322; .NET4.0C; .NET4.0E; InfoPath.3)

Host: 90[.]160[.]138[.]175

Neither fields represent ideal candidates for signature generation.

Other types of C2 packets can be reliably identified with traffic patterns. Such reliable patterns are typically character strings that uniquely identify a certain type of C2 traffic and are not found in other (i.e. benign) traffic sessions. Still, this approach has the same disadvantage as IP/domain name-based detection due to the inherent challenges of maintaining an up-to-date and complete set of traffic patterns.

Due to these shortcomings, the application of machine learning is imperative to achieve flexible and reliable detection of C2 traffic. This is critical for detecting novel types of network-based attacks.

C2 Detection with Deep Learning

It is crucial to detect these malicious C2 traffic sessions promptly. As mentioned above, this is traditionally done through the usage of static signatures on payloads and URLs. However, these signatures are not exhaustive and can not detect novel C2 sessions. For these reasons, we researched a deep learning model that can automatically extract the important features from a vast amount of data to detect malicious C2 sessions.

Our deep learning model leverages advanced machine learning algorithms to learn the content and context from a network session and determine if it connects to a malicious C2 server. Our detection module determines the probability of the session being malicious. Based on the predetermined threshold, we can classify if a given session is malicious or not. For this blog, we tested a model trained on ~60 million HTTP session headers with ~36 million benign and ~24 million malicious sessions. This dataset was collected in 2019.

The hyperparameters for training the deep learning model are computed so the false positive rate of the model remains below 0.025%. We tested this model for over four months and observed that the average false positive rate remained below 0.02% with more than 98% precision.

How Does a Deep Learning Model Detect the Traffic Packets Shown Above?

Deep learning models extract features implicitly from the training data. Thus, it may not be possible to ascertain precisely which feature or sequence of features from a packet header triggers detection.

Deep neural networks have many parameters to obtain a highly expressive data representation compared to traditional statistical models. By presenting a deep learning model with millions of known malicious data packets, a neural network is trained to recognize the general structure of a C2 traffic packet. Consequently, the factors involved in packet classification do not depend on a single field (such as the host name), but on various features -- the combination of characters and words or the structure of the packet. The features that distinguish a benign packet from a malicious one are automatically recognized by the neural network during the training process of millions of labeled data points.

To better understand how a detection decision is made, we re-create the packet headers by removing some critical information (e.g., hostname, URI paths) and evaluate these re-created headers with our deep learning model. We summarized our results for different types of malicious C2 traffic in Table 1 and Table 2.

In Table 1, we present the probability with which our model could detect a session as malicious by testing the full HTTP header, as well as the header without the uri-path, hostname, user-agent or referer. These context-fields were removed one at a time. We observed that the model could detect all four C2 sessions with high confidence

when all header field information was present. We also see that the model was not reliant on one specific context-field and could detect malicious C2 detection without some of them present.

Malware C2	Full header	Without uri-path	Without hostname	Without user-agent	Without referer
Emotet C2 traffic 1	99.72	99.86	96.28	97.37	99.55
Emotet C2 traffic 2	99.79	99.91	98.04	95.48	99.76
Sality C2 traffic 1	99.99	99.99	99.98	99.99	NA
Sality C2 traffic 2	99.99	99.99	99.98	99.99	NA

Table 1. Performance of our deep learning model on session headers with different levels of information (the values in the table show the model confidence of a session being malicious; NA = the packet didn't have this field).

As we discussed above, a significant challenge in detecting C2 traffic with static signatures is differentiating reliable patterns from the network traffic. But, this is not the only pitfall signatures face in the detection of C2 traffic. A slight modification of C2 malware traffic could render a signature ineffective. Consider the Sality C2 packet shown in Figure 1. The pattern 'GET /sobaka1.gif' is a potential candidate to be used in a signature in an IPS. An advanced malware may frequently change the command pattern in its traffic payload to bypass packet inspection by an IPS.

We simulate such behavior by modifying packet headers and analyze how the detection output of our deep learning model changes. Consider the example below. We changed the uri-path in the Sality C2 packet shown in Figure 1 from 'sobaka1.gif?12db3cf=98861835' to '/nobata2.gif?52ad3pf=77952613' and our model was still able to detect the packet header as malicious with 99.9% confidence. The full modified packet header is shown in Figure 5.

```
GET /nobata2.gif?52ad3pf=77952613 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0b; Windows NT 6.0)
Host: padrup[.]com
Cache-Control: no-cache
Cookie: jsessionid=85b50d8fab658ecb9f79aa4de6039c87
```

Figure 5. Modified packet header of Sality C2 traffic.

In Table 2, we present the prediction results of our model on changing values of different context fields, one at a time, in the request header. We observed that our model could detect the C2 sessions with modified values in the

context-fields. This shows that our deep learning model is not relying on one specific context-field value, but is learning from the overall structure of the request header.

Malware C2	Full header	Modified uripath	Modified host	Modified user-agent	Modified referer
Emotet C2 traffic 1	99.72	99.72	98.60	98.86	99.63
Emotet C2 traffic 2	99.79	99.92	99.94	99.99	99.94
Sality C2 traffic 1	99.99	99.99	99.96	99.95	NA
Sality C2 traffic 2	99.99	99.99	99.85	99.97	NA

Table 2. Performance of our deep learning model on session headers with changes made on different context-fields of a request header (the values in the table shows the model confidence of a session being malicious; NA = the packet didn't have this field).

In addition to specific payload patterns in packet headers, the ordering of fields can play a role in detecting C2 sessions with deep learning. In Table 3, we show how our model performs when the packets arrive in different context-field structures. For evaluation, we organized the context fields -- host name (H), referer (R) for Emotet cache-control (C) for Sality and user-agent (UA) in different arrangements. We found that our deep learning model could identify the payloads correctly even when the packet structure was changed.

Malware C2	Full header	(UP|H|R/C|UA)	(UP|H|UA|R/C)	(UP|UA|H|R/C)
Emotet C2 traffic 1	99.72	99.65	98.84	99.50
Emotet C2 traffic 2	99.79	98.27	99.51	99.74
Sality C2 traffic 1	99.99	99.90	99.99	99.99
Sality C2 traffic 2	99.99	99.93	99.98	99.98

Table 3. Performance of our deep learning model on session headers with reordered context-fields (H = Host name, UA = User-Agent, UP = Uri-Path, R = Referer/C=Cache-Control) (the values in the table shows the model confidence of a session being malicious; NA = the packet didn't have this field).

Overall, the results presented above demonstrate the strength of our deep learning model on detecting the malicious C2 sessions of the malware families like Emotet and Sality. These malware families might connect to different host servers and transfer additional information to conduct attacks, making it challenging to capture them with static signatures.

Conclusion

Our research on using deep learning for C2 traffic detection shows the potential and necessity to use advanced machine learning for intrusion detection and prevention. For novel attacks and zero-day vulnerabilities, it is critical to rely on systems that can identify attacks based on known traffic features and identify unknown types of malicious network traffic to detect and prevent advanced threat campaigns at an early stage. The results Unit 42 sees in various research projects directly contribute to Threat Prevention and WildFire security subscriptions to ensure the protection of our Next-Generation Firewall customers.
