Observations about the attack on WikiLeaks

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 1 September 2017

http://www.bortzmeyer.org/observations-wikileaks.html

On 30 August, this year, a technical attack was performed against WikiLeaks, leading some visitors of WikiLeaks' Web site to see instead a claim by "OurMine" that they seized control of WikiLeaks' servers. A lot of stupid things, by ignorant people (both WikiLeaks fans and enemies), have been said on the networks, about this attack. Most of the time, they did not bother to check any facts, and they did not demonstrate any knowledge of the technical infrastructure. Here, I want to describe the bare facts, as seen from technical observations. Spoiler : I have no sensational revelations to make.

First, the basic fact : some people saw something which was obviously not WikiLeaks' Web site : screenshots of the page are here <https://www.theguardian.com/technology/2017/aug/31/wikileaks-hacked-ourmine-group-julian-assange-dns-attack> or here <https://www.reddit.com/r/conspiracy/comments/6x4hpr/wikileaks_dns_hijacked_by_ourmine_hacking_group/>. Some people deduced from that that WikiLeaks' Web server was cracked and the crackers modified its content (you can find this in The Verge <https://www.theverge.com/2017/8/31/16232164/wikileaks-hacked-ourmine-website> for instance). That was a bold deduction : the complete process from the user's browser to the display of the Web page is quite complicated, with a lot of actors, and many things can go wrong.

In the case of WikiLeaks, it appeared rapidly that the Web server was not cracked but that the attack targeted successfully the `wikileaks.org` domain name. Observations (more on that later) show that the name `wikileaks.org` was not resolved into the usual IP address but in another one, located in a different hoster. How is it possible ? What are the consequences ?

You should remember that investigation of digital incidents on the Internet is difficult. The external analyst does not have all the data. Sometimes, when the analysis starts, it is too late, the data already changed. And the internal analyst almost never publishes everything, and sometimes even lies. There are some organisations that are more open in their communication (see this Cloudflare report <https://blog.cloudflare.com/how-and-why-the-leap-second-affected-cloudflare-dns/> or this Gandi report <https://news.gandi.net/en/2017/07/detailed-incident-report/>) but they are the exceptions rather than the rule. Here, WikiLeaks reacted like the typical corporation, denying the problem <https://twitter.com/wikileaks/status/903225530635485185>, then trying to downplay it <https://twitter.com/JulianAssange/status/903217338593497088>,

and not publishing anything <https://wikileaks.org/-News-.html> of value for the users. So, most of the claims that you can read about network incidents are not backed by facts, specially not publicly-verifiable facts. The problem is obviously worse in that case, because WikiLeaks is at the center of many hot controversies. For instance, some WikiLeaks fans claimed from the beginning "WikiLeaks' servers have not been compromised" while they had zero actual information, and, anyway, not enough time to analyze it.

So, the issue was with the domain name `wikileaks.org`. To explain what happened, we need to go back to the DNS, both a critical infrastructure of the Internet, and a widely unknown (or underknown) technology. The DNS is a database indexed by domain names (like `wikileaks.org` or `ssi.gouv.fr`). When you query the DNS for a given domain name, you get various technical informations such as IP addresses of servers, cryptographic keys, name of servers, etc. When the typical Web browser goes to `http://www.okstate.com/`, the software on the user's machine performs a DNS query for the name `www.okstate.com`, and gets back the IP address of the HTTP server. It then connects to the server.

From this very short description, you can see that s[Caractère Unicode non montré [1]]he who controls the DNS controls **where** the user will eventually go and what s[Caractère Unicode non montré]he will see. And the entire **DNS resolution** process (from a name to the data) is itself quite complicated, offering many opportunities for an attacker. Summary so far : DNS is critical, and most organisations underestimate it (or, worse, claim it is not their responsability).

And where do the data in the DNS come from ? That's the biggest source of vulnerabilities : unlike what many people said, most so-called "DNS attacks" are not DNS attacks at all, meaning they don't exploit a weakness in the DNS protocol. Most of the time, they are attacks against the **provisioning** infrastructure, the set of actors and servers that domain name holders (such as WikiLeaks for `wikileaks.org`) use to provision the data. Let's say you are Pat Smith, responsible for the online activity of an organisation named the Foobar Society. You have the domain name `foobar.example`. The Web site is hosted at Wonderful Hosting, Inc. After you've choosen a TLD (and I recommend you read the excellent EFF survey <https://www.eff.org/wp/which-internet-registries-offer-best-protect before you do so), you'll typically need to choose a registrar which will act as a proxy between you and the actual registry of the TLD (here, the fictitious `.example`). Most of the time, you, Pat Smith, will connect to the Web site of the registrar, create an account, and configure the data which will ultimately appear in the DNS. For instance, when the Web site is created at Wonderful Hosting, Pat will enter its IP address in the control panel provided by the registrar. You can see immediately that this required Pat to log in the said control panel. If Pat used a weak password, or wrote it down under h[Caractère Unicode non montré]is[Caractère Unicode non montré]er desk or if Pat is gullible and believes a phone call asking h[Caractère Unicode non montré]im[Caractère Unicode non montré]er to give the password, the account may be compromised, and the attacker may log in instead of Pat and put the IP address of h[Caractère Unicode non montré]er[Caractère Unicode non montré]is choosing. This kind of attacks is very common, and illustrate the fact that not all attacks are technically complicated.

So, what happened in the WikiLeaks case ? (Warning, it will now become more technical.) We'll first use a "passive DNS" base, DNSDB <https://www.dnsdb.info/>. This sort of databases observes the DNS traffic (which is most of the time in clear, see RFC 7626 [2]) and record it, allowing its users to time-travel. DNSDB is not public, I'm sorry, so for this one, you'll have to trust me. (That's why real-time reaction is important : when you arrive too late, the only tools to observe an attack are specialized tools like this one.) What's in DNSDB ?

1. Car trop difficile à faire afficher par LaTeX
2. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc7626.txt`

```
;;  bailiwick: org.
;;      count: 9
;; first seen: 2017-08-30 04:28:40 -0000
;;  last seen: 2017-08-30 04:30:28 -0000
wikileaks.org. IN NS ns1.rivalhost.com.
wikileaks.org. IN NS ns2.rivalhost.com.
wikileaks.org. IN NS ns3.rivalhost.com.

;;  bailiwick: org.
;;      count: 474
;; first seen: 2017-08-30 04:20:15 -0000
;;  last seen: 2017-08-30 04:28:41 -0000
wikileaks.org. IN NS ns1.rival-dns.com.
wikileaks.org. IN NS ns2.rival-dns.com.
wikileaks.org. IN NS ns3.rival-dns.com.
```

What does it mean ? That during the attack (around 04 :30 UTC), the `.org` registry was replying with the illegitimate set of servers. The usual servers are (we use the dig tool, the best tool to debug DNS issues) :

```
% dig @a0.org.afilias-nst.info. NS wikileaks.org
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21194
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 3
...
;; AUTHORITY SECTION:
wikileaks.org. 86400 IN NS ns1.wikileaks.org.
wikileaks.org. 86400 IN NS ns2.wikileaks.org.

;; ADDITIONAL SECTION:
ns1.wikileaks.org. 86400 IN A 46.28.206.81
ns2.wikileaks.org. 86400 IN A 46.28.206.82
...
;; SERVER: 2001:500:e::1#53(2001:500:e::1)
;; WHEN: Fri Sep 01 09:18:14 CEST 2017
...
```

(And, yes, there is a discrepancy between what is served by the registry and what's inside `nsX.wikileaks.org` name servers : whoever manages WikiLeaks DNS does a sloppy job. That's why it is often useful to query the parent's name servers, like I did here.)

So, the name servers were changed, for rogue ones. Note there was also a discrepancy during the attack. These rogue servers gave a different set of NS (Name Servers), according to DNSDB :

```
;;  bailiwick: wikileaks.org.
;;      count: 1
;; first seen: 2017-08-31 02:02:38 -0000
;;  last seen: 2017-08-31 02:02:38 -0000
wikileaks.org. IN NS ns1.rivalhost-global-dns.com.
wikileaks.org. IN NS ns2.rivalhost-global-dns.com.
```

Note that it does not mean that the DNS hoster of the attacker, Rival <`https://www.rivalhost.com/`>, is an accomplice. They may simply have a rogue customer. Any big service provider will have some rotten apples among its clients.

You can see the date of the last change in whois output, when everything was put back in place :

```
% whois wikileaks.org
...
Updated Date: 2017-08-31T15:01:04Z
```

Surely enough, the rogue name servers were serving IP addresses pointing to the "false" Web site. Again, in DNSDB :

```
;;  bailiwick: wikileaks.org.
;;      count: 44
;; first seen: 2017-08-30 04:29:07 -0000
;;  last seen: 2017-08-31 07:22:05 -0000
wikileaks.org. IN A 181.215.237.148
```

The normal IP addresses of WikiLeaks are in the prefixes `95.211.113.XXX`, `141.105.XXX` and `195.35.109.XXX` (`dig A wikileaks.org` if you want to see them, or use a DNS Looking Glass <`https://dns.bortzmeyer.org/wikileaks.org`>). `181.215.237.148` is the address of the rogue Web site, hosted by Rival again, as can be seen with the whois tool :

```
% whois 181.215.237.148
inetnum:     181.215.236/23
status:      reallocated
owner:       Digital Energy Technologies Chile SpA
ownerid:     US-DETC5-LACNIC
responsible: RivalHost, LLC.
address:     Waterwood Parkway, 1015, Suite G, C-1
address:     73034 - Edmond - OK
country:     US
owner-c:     VIG28
tech-c:      VIG28
abuse-c:     VIG28
...
nic-hdl:     VIG28
person:      AS61440 Network Operating Center
e-mail:      noc@AS61440.NET
address:     Moneda, 970, Piso 5
address:     8320313 - Santiago - RM
country:     CL
```

(It also shows that this prefix was allocated in Chile, the world is a complicated place, and the Internet even more so.)

So, this was the modus operandi of the cracker. S[Caractère Unicode non montré]he managed to change the set of name servers serving `wikileaks.org`, and that gave h[Caractère Unicode non montré]im[Caractère Unicode non montré]er the ability to send visitors to a server s[Caractère Unicode non montré]he controlled. (Note that this HTTP server, `181.215.237.148`, no longer serves the cracker's page : it was probably removed by the provider.)

Many people on the social networks claimed that the attack was done by "DNS poisoning". First, a word of warning by a DNS professional : when someone types "DNS poisoning", you can be pretty sure s[Caractère Unicode non montré]he knows next to nothing about DNS. DNS poisoning is a very specific attack, for which we have solutions (DNSSEC, mentioned later), but it does not seem to be very common (read again my warning at the beginning : most attacks are never properly analyzed and documented, so it is hard to be more precise). What **is** very common are attacks against the domain name

provisioning system. This is, for instance, what happened to the New York Times in 2013, from an attack by the infamous SEA (see NYT paper <http://www.nytimes.com/2013/08/28/business/media/hacking-attack-is-suspected-on-times-web-site.html> and a technical analysis <https://blog.cloudflare.com/details-behind-todays-internet-hacks/>). More recently, there was the attack against St. Louis Federal Reserve <https://krebsonsecurity.com/2015/05/st-louis-federal-reserve-suffers-dns-breach/> and many others. These attacks **don't** use the DNS protocol and it is quite a stretch to label them as "DNS attacks" or, even worse, "DNS poisoning".

What are the consequences of such an attack? As explained earlier, once you control the DNS, you control everything. You can redirect users to a Web site (not only the external visitors, but also the employees of the targeted organisation, when they connect to internal services, potentially stealing passwords and other informations), hijack the emails, etc. So, claiming that "the servers were not compromised" (technically true) is almost useless. With an attack on domain names, the cracker does not need to compromise the servers.

Who was cracked in the WikiLeaks case? From the outside, we can say with confidence that the name servers were changed. The weakness could have been at the holder (WikiLeaks), at the registrar (Dynadot <https://www.dynadot.com/>, an information you also get with whois), or at the registry (.org, administratively managed by PIR and technically by Afilias). From the information available, one cannot say where the problem was (so, people who publicly shouted that "WikiLeaks is not responsible" were showing their blind faith, not their analytic abilities). Of course, most of the times, the weakest link is the user (weak password to the registrar portal, and not activating 2FA), but some registrars or registries displayed in the past serious security weaknesses. The only thing we can say is that no other domain name appeared to have been hijacked. (When someone takes control of a registrar or registry, s[Caractère Unicode non montré]he can change many domain names.)

I said before that, when you control a domain name, you can send both external and internal visitors to the server you want. That was not entirely true, since good security relies on defence in depth and some measures can be taken to limit the risk, even if your domain name is compromised. One of them is of course having HTTPS (it is the case of WikiLeaks), with redirection from the plain HTTP site, and HSTS (standardized in RFC 6797), to avoid that regular visitors go through the insecure HTTP. Again, WikiLeaks use it :

```
%  wget --server-response --output-document /dev/null https://wikileaks.org/
...
Strict-Transport-Security: max-age=25920000; includeSubDomains; preload
```

These techniques will at least raise an alarm, telling the visitor that something is fishy. (There is also HPKP - RFC 7649 - but it does not seem deployed by Wikileaks ; it should be noticed it is more risky.)

In the same way, using Tor to go to a .onion URL would also help. But I have not been able to find a .onion for WikiLeaks (the http://suw74isz7wqzpmgu.onion/ indicated on the wiki <https://www.wikileaks.org/wiki/WikiLeaks:Tor> does not work, the http://wlupld3ptjvsgwqw.onion seems to be just for uploading).

One can also limit the risk coming from an account compromise by enabling registry lock, a technique offered by most TLD (including .org) to prevent unauthorized changes. When activated, it requires extra steps and checking for any change. I cannot say, from the outside, if WikiLeaks enabled it but sensitive domain names **must** do it.

Funny enough, with so many people claiming it was "DNS poisoning", the best protection against this specific attack, DNSSEC, is **not** enabled by WikiLeaks (there is a DNSSEC key in `wikileaks.org` but no signatures and no DS record in the parent). If `wikileaks.org` was signed, and if you use a validating DNS resolver (everybody should), you cannot fall for a DNS poisoning attack against WikiLeaks. Of course, if the attack is, instead, a compromise of holder account, registrar or registry, DNSSEC would not help a lot.

A bit of technical fun at the end. WikiLeaks uses glue records for its name servers. They are nameserver names which are under the domain they serve, thus creating a chicken-and-egg problem. To allow the DNS client to query them, the parent has to know the IP address of this name server. This is what is called a glue record. DNSDB shows us that the glue for `ns1.wikileaks.org` was apparently modified (note that it was several hours after the main attack) :

```
;;  bailiwick: org.
;;      count: 546
;; first seen: 2017-08-31 00:23:13 -0000
;;  last seen: 2017-08-31 06:22:42 -0000
ns1.wikileaks.org. IN A 191.101.26.67
```

This machine is still up and serves a funny value for `wikileaks.org` (again, you can use a DNS Looking Glass <https://dns.bortzmeyer.org/wikileaks.org?server=191.101.26.67>) :

```
% dig @191.101.26.67 A wikileaks.org
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53887
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
wikileaks.org. 400 IN A 127.0.0.1
```

This IP address, meaning `localhost`, was indeed seen by some DNSDB sensors :

```
;;  bailiwick: wikileaks.org.
;;      count: 1
;; first seen: 2017-08-31 09:17:29 -0000
;;  last seen: 2017-08-31 09:17:29 -0000
wikileaks.org. IN A 127.0.0.1
```

Since the DNS heavily relies on caching, the information was still seen even after the configuration was fixed. Here, we use the RIPE Atlas probes <https://atlas.ripe.net/> with the atlas-resolve <https://labs.ripe.net/Members/stephane_bortzmeyer/using-ripe-atlas-to-debug-network-c tool to see how many probes still saw the wrong value (pay attention to the date and time, all in UTC, which is the rule when analyzing Internet problems) :

```
% atlas-resolve -r 1000 -t A wikileaks.org
[141.105.65.113 141.105.69.239 195.35.109.44 195.35.109.53 95.211.113.131 95.211.113.154] : 850 occurrences
[195.175.254.2] : 2 occurrences
[127.0.0.1] : 126 occurrences
Test #9261634 done at 2017-08-31T10:03:24Z
```