Fast Flux 101: How Cybercriminals Improve the Resilience of Their Infrastructure to Evade Detection and Law Enforcement Takedowns

Executive Summary

Fast flux is a technique used by cybercriminals to increase their infrastructure's resilience by making law enforcement takedown of their servers and denylisting of their IP addresses harder. It is critical for these cybercriminals to maintain their networks' uptime to avoid losses to their revenue streams, including phishing and scam campaigns, botnet rental and illegal gambling operations.

The motivation for cybercriminals to build fast flux networks is similar to that of benign service providers, who build redundancy in their systems to ensure uptime, for example, by utilizing Round Robin in the Domain Name System (RRDNS) or Content Delivery Networks (CDNs). The main difference is that fast flux networks are used to enable illegal and malicious activities. Therefore, operators need to rely on peculiar techniques such as frequently changing their IP addresses and using botnets or bulletproof hosting (hosting providers who tend not to respond to takedown requests). A fast flux network is "fast" because, using DNS, it quickly rotates through many bots, using each one for only a short time to make IP-based denylisting and takedown efforts difficult.

In this blog, we provide a fictional scenario of a cat-and-mouse game between cybercriminals and law enforcement. We illustrate how cybercriminals use single fast flux networks and more advanced techniques such as double flux (when the domain name resolution becomes part of the fast flux network) and Domain Generation Algorithms (DGAs) to hamper domain denylisting and takedown efforts.

Additionally, we cover three case studies that show the wide range of malicious activities that fast fluxing can be used for. We observe scammers using fast flux domains to operate social engineering

pages in many different languages, cybercriminals infecting machines with Smoke Loader malware and using fast fluxing for their command and control (C2) domains and finally, we show how fast flux domains are used to operate illicit adult and gambling sites.

Palo Alto Networks provides protection against fast flux and DGA domains leveraging our classifiers in multiple Palo Alto Networks Next-Generation Firewall security subscriptions, including URL Filtering and DNS Security.

Fast Flux Fictional Scenario

Fast flux networks can be used to support a wide variety of criminal endeavors, such as phishing, scams, malware distribution and botnet operations. The term "fast flux" originates from April Lorenzen, who observed early use of this technique. To explain how fast flux works and to provide background, we will start with a fictitious example of a phishing operator, Mallory. Mallory would like to harvest user credentials that he can later sell on various illicit underground markets.

- **Mallory**: The villain involved in phishing.
- **Bart**: Mallory's friend who compromises machines and builds botnets.
- **Alice**: Unsuspecting customer of Rainbow Bank.
- **Emilia**: The protagonist and the leader of a law enforcement team tasked with stopping the phishing attacks launched lately against Rainbow Bank's customers.

Before his venture into fast fluxing, Mallory learned that the domain name system translates domain names that are easy to remember for humans (e.g., paloaltonetworks.com) to IP addresses (e.g., 34.107.151.202) understood by machines. These IP addresses are what machines use on the Internet to find each other and to be able to communicate.

Mallory started off by creating a website mimicking a known bank, rainbowbank[.]com. He set up a fake bank website called ra**l**nbowbank[.]com on a rented host with IP address 10.123.34[.]55.

Mallory chose the typosquatting domain ra**l**nbowbank[.]com as it looks very similar to the bank's real domain name. In general, typosquatting domain names (ra**l**nbowbank[.]com) are misspelled variants of target domain names (rainbowbank[.]com), registered to profit from users' typing mistakes or deceive users into believing that they are the correct target domain. Our blog on cybersquatting details the

dangers of such squatting domains and how Palo Alto Networks protects its customers against them.

Mallory paid his friend Bart to send out carefully crafted spam emails to Rainbow Bank's customers. These emails notified the bank's customers that their account credentials need to be validated by clicking on the email link. The link in the emails redirected users to the phishing page hosted on Mallory's server. Thus, Mallory was able to start collecting credentials. Fortunately, his site was reported, and law enforcement agent Emilia quickly reacted to take down the server hosting Mallory's website.

But Mallory had seen how lucrative phishing could be. He realized that next time, he just needed to make sure Emilia couldn't take down his website so easily. When setting up ralnbowbank[.]com, Mallory learned how he could set the A record for his phishing domain, which holds the IP address of the server he uses.

DOMAIN NAME	TTL	TYPE	RECORD
RALNBOWBANK.COM.	14400	A	10.123.34.55

Mallory also heard about fast flux networks on his favorite forum, hack-a-rainbow[.]com, and he wanted to give it a try. He read that the most important part of the technique was for him to get a lot of compromised machines, so he can rapidly switch the DNS records.

First, Mallory wrote a script that can automatically update the hosts where the DNS records of ral nbowbank[.]com are pointed. Below is an example of how the IP addresses configured by Mallory on the DNS server might have looked at the start of his phishing campaign.

DOMAIN NAME	TTL	TYPE	RECORD
RALNBOWBANK.COM.	160	A	101.14.66.2
RALNBOWBANK.COM.	160	A	222.14.10.4
RALNBOWBANK.COM.	160	A	23.124.228.102
RALNBOWBANK.COM.	160	A	101.14.66.22

His script frequently updated the DNS records to avoid detection, as shown in the example below. Note that the Time-To-Live (TTL) value was set to a small number to ensure earlier DNS responses were not cached for long.

DOMAIN NAME	TTL	TYPE	RECORD
RALNBOWBANK.COM.	160	A	181.214.153.22
RALNBOWBANK.COM.	160	A	96.44.162.82
RALNBOWBANK.COM.	160	A	106.253.253.5
RALNBOWBANK.COM.	160	A	139.99.91.95

Figure 1 depicts how the fast flux architecture Mallory designed looked. Steps 1 and 2 show the DNS resolution part controlled by Mallory. Furthermore, Mallory knew that Bart uses a botnet to send out spam emails, so he asked him if he could spare a few thousand bots, offering to rent them. Bart thought, *Why not?* He rented out these bots to some of his other friends sometimes, too. Bart agreed for a few hundred dollars to allow Mallory to install a proxy script on these bots. The installed proxy script relays requests between the bots and Mallory's main server, which he likes to call the Mothership. Steps 3-6 on Figure 1 show how the HTTP requests are proxied through the botnet. Based on the IP address returned in the DNS response, the bot used for proxying changed frequently.

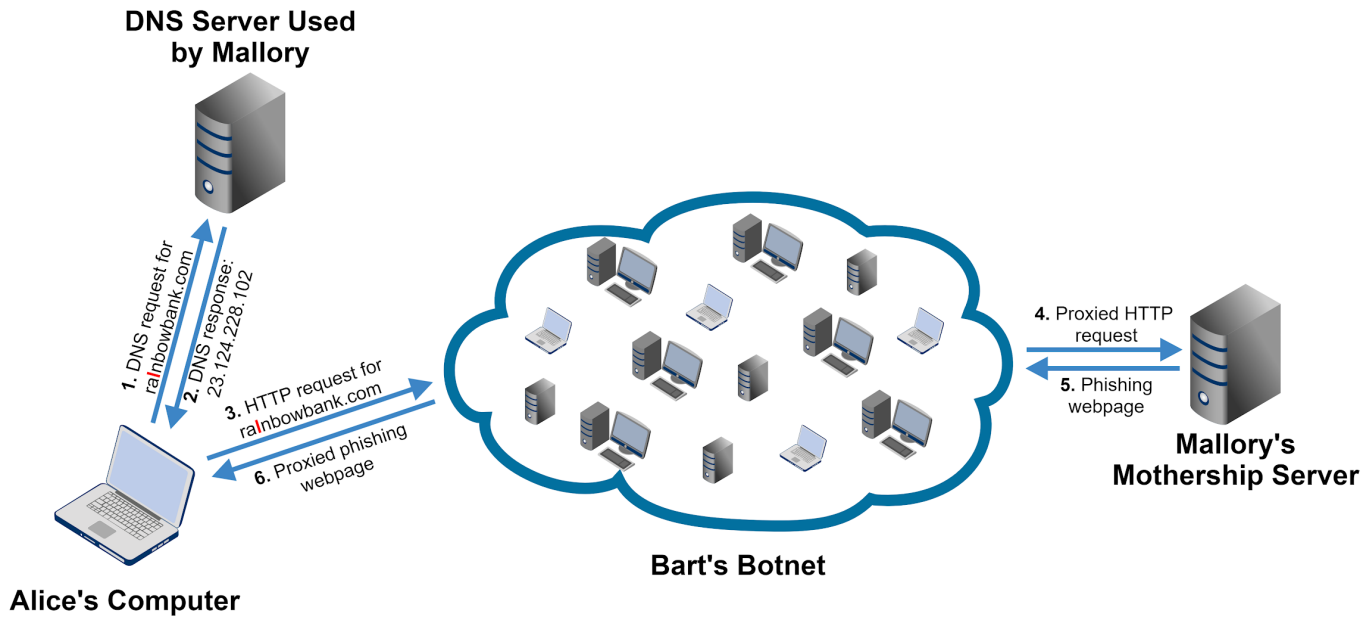

Figure 1. Mallory's fast flux architecture.

Using a fast flux network was a great success for Mallory, and it was much harder for Emilia and her team to clean up his campaign this time. Next, we will discuss further hurdles Mallory must overcome to run his malicious campaign successfully and how he can leverage more advanced architectures to make his phishing campaign even more resilient to takedowns and denylists.

Advanced Techniques

This section discusses three techniques and tactics Mallory and Bart can leverage to improve their infrastructure further.

Double Flux

Emilia realized that it became untenable for her team to shut down all the hosting servers in Mallory's fast flux network. Luckily, Emilia had an idea. If the team cannot go after the hosts, then they should take down the DNS server that provides DNS resolution for Mallory. After Emilia successfully curbed Mallory's attack a second time, Mallory thought, *Let me apply a quick fix and move the DNS resolution itself to a fast flux network.*

To understand how Mallory's new architecture worked, let us first consider how hierarchical DNS resolution looks using Figure 2. When Alice's computer queries ra**l**nbowbank[.]com, it first needs to query the DNS root servers to find the .com Top-Level Domain (TLD) name server, which knows where Mallory's name server is. Second, it queries the .com TLD's name server to reach Mallory's name server, which can finally return the IP address for ra**l**nbowbank[.]com.

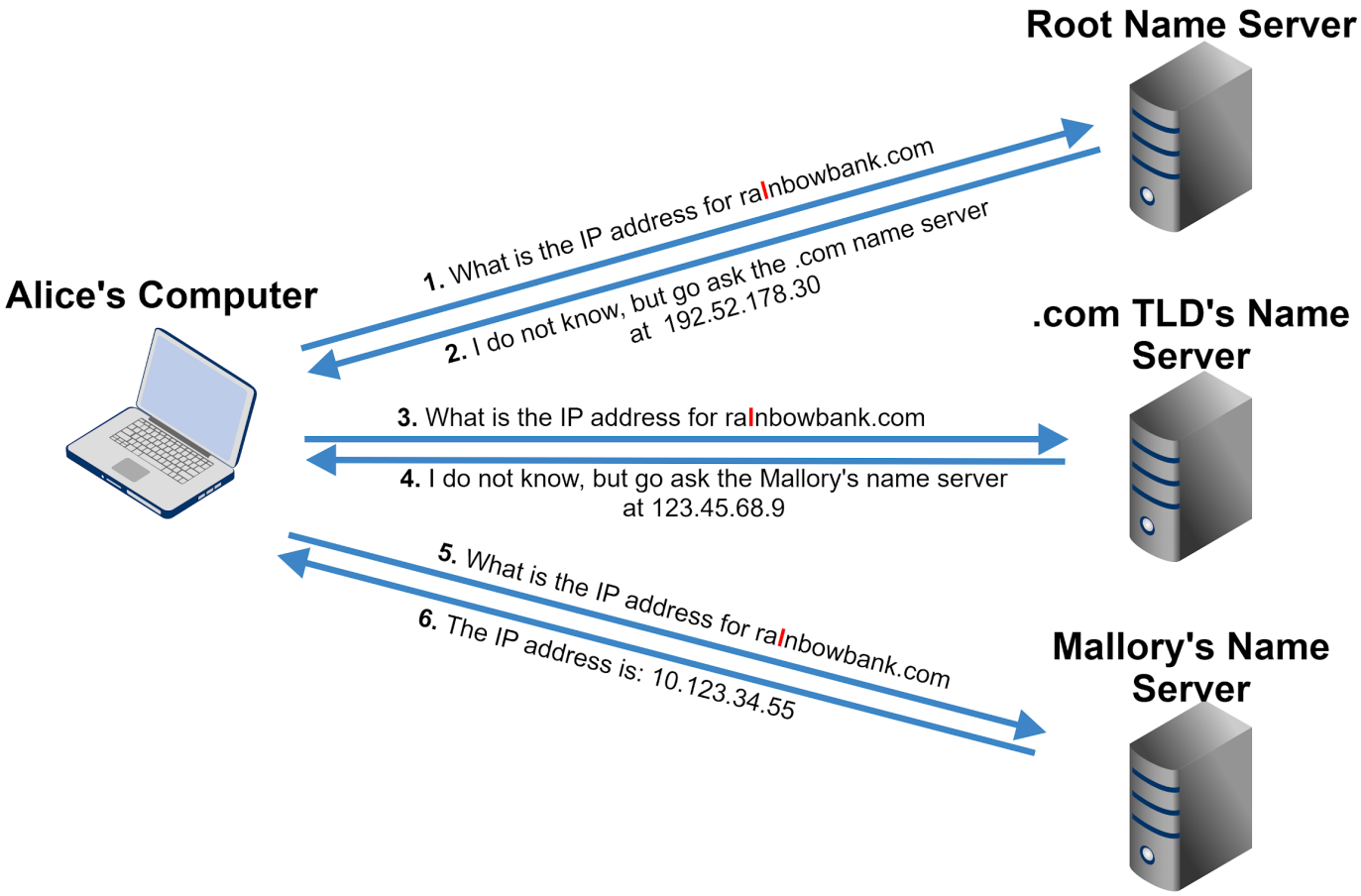

Figure 2. Example of how DNS resolution works.

Mallory, armed with the understanding of how DNS works, set up a fast flux network to replace his previous name server. This setup is called a double flux network, as shown in Figure 3. He was also able to start rapidly changing the IP address of ns1.ralnbowbank[.]com and ns2.ralnbowbank[.]com, the name servers of ralnbowbank[.]com.

Emilia's team was in trouble again. It seemed like their approach of taking down hosts did not work anymore. Similarly, IP denylists became ineffective against Mallory's double flux network. What did Emilia think of next?

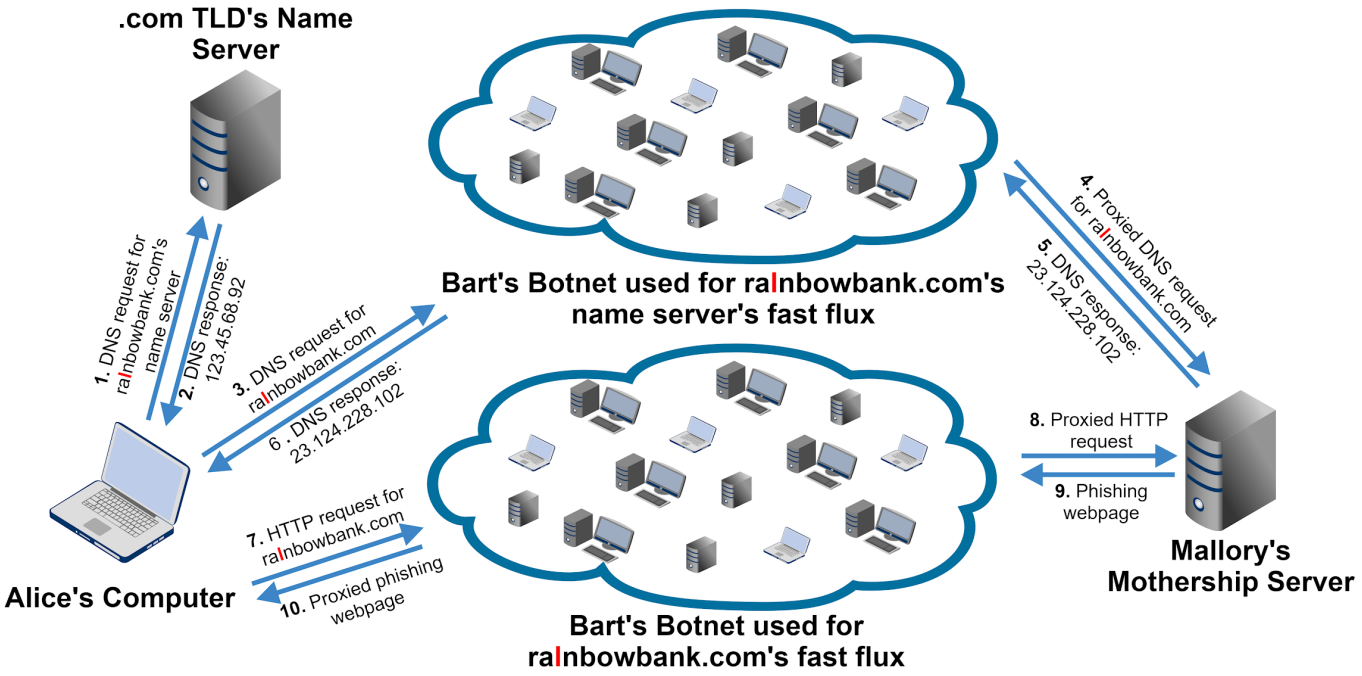

Figure 3. Mallory's double flux architecture.

The Domain Wars

Emilia was only left with one option. She had to go after the domain name itself, ralnbowbank[.]com. Emilia had a few options to consider. She could file a complaint with the operator of the .com TLD (TLD operators are called registries) managing the name server (NS) records of ralnbowbank[.]com or with the reseller (registrar) who sold the domain name to Mallory – either complaint could ask to have the malicious domain removed. It would also be possible to use the Uniform Domain Name Dispute Resolution Policy (referred to as UDRP) or Uniform Rapid Suspension (URS). However, these processes are significantly slower compared to a responsive and responsible registry or registrar.

The domain wars started when Emilia had ralnbowbank[.]com taken down with the registrar Penny Domains. Mallory, now a seasoned cybercriminal, was quick to register new domains, each used for a short period of time. Mallory's job was made easier by all the new TLDs (e.g., .xyz), where he was able to acquire domains for cheap and without identity verification. Meanwhile, Emilia's team also gained the expertise to find Mallory's domains faster and had better processes in place to have these domains removed. Still, Emilia's team fell behind and could not keep up with Mallory.

Finally, Emilia caught Mallory using detective work. Mallory always communicated with Bart on the same forum, where his username was similar to his social media username. Using this and some other clues derived from Mallory's operation, Emilia's team identified who Mallory really was, thus ending his phishing operations for good.

Domain Generation Algorithms

Unfortunately, Mallory was not the only perpetrator of malicious activity who used Bart's botnet. After Mallory was prosecuted, Bart used his botnet to conduct a revenge Distributed Denial of Service (DDOS) attack against Emilia's department. Further, Bart had many friends who used his botnet for different malicious activities such as scams, malware distribution, distributions of potentially unwanted programs (PUPs), illegal gambling operations and phishing.

Next, Emilia's team went after Bart's botnet. Figure 4 provides an overview of Bart's botnet infrastructure. Bart was not picky when it came to compromising machines for his botnet, and he had anything from weaker personal computers to powerful servers under his control. On each bot, he installed malware that was able to ask his command and control (C2) servers for further code to run. When a bot wanted to communicate with the C2 server, the malware first tried to resolve the domain bartrules247[.]net to get the C2's IP address.

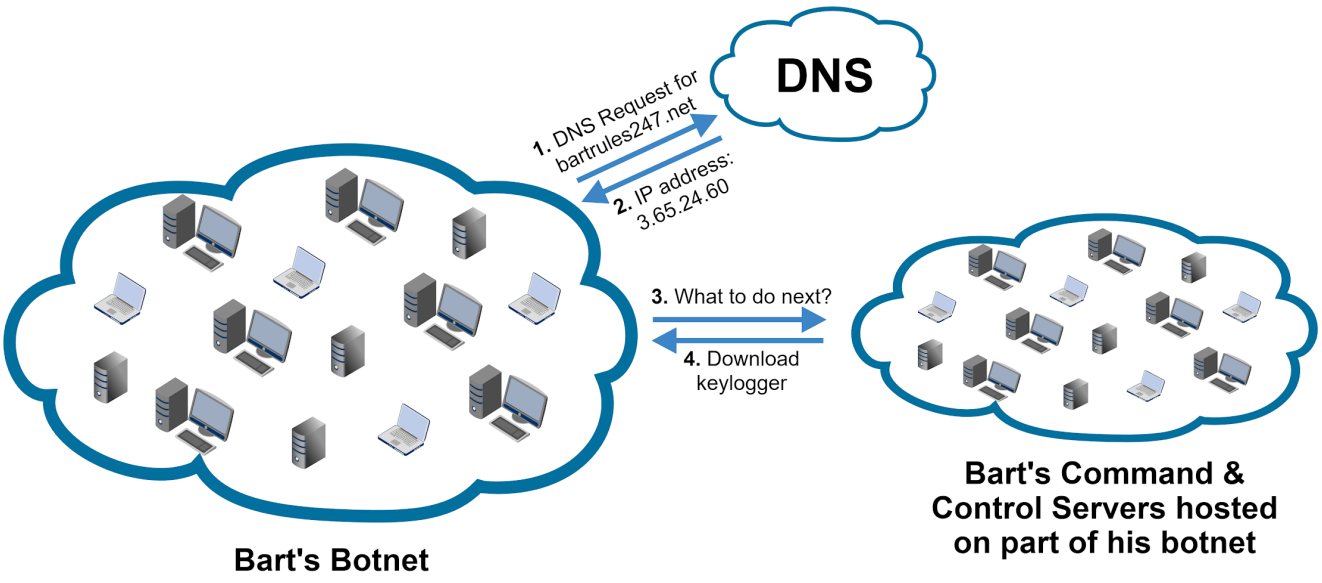

Figure 4. Bart's botnet infrastructure.

When Emilia reverse engineered Bart’s code (which she found on a compromised machine), she learned that it used the domain name bartrules247[.]net. When her team took down Bart’s domain name, the botnet went offline.

To counter Emilia’s effort, Bart devised a tricky strategy. He hid a piece of code in his malware codebase that generated a different set of thousand domain names every day. Bart’s bot tried to access all of these domains, and if even one of them was registered by Bart, the communication succeeded. Bart’s new approach made it impossible for Emilia to take down his botnet, as there would have been millions of potential domain names Emilia would have needed to take control of over time.

In the end, Bart was caught just like Mallory, thanks to Emilia’s and her team’s hard work. The team discovered where Bart lived because Mallory had sent him a birthday present the previous year. While Bart and Mallory's operational mistakes might seem like rookie blunders, they represent typical ways for criminals to get caught.

The example scenario presented in this blog is simplified to showcase some of the ingenious ways cybercriminals try to harden their infrastructure against law enforcement and hide from security researchers.

Detecting Fast Flux and DGA Domains

In our example, we have seen how challenging it was for Emilia's team to take down fast flux networks. Palo Alto Networks Next-Generation Firewall security subscriptions URL Filtering and DNS Security include an automated classifier to detect fast flux domains and protect users.

Our classifier builds on two distinctive features of these domain names. First, fast flux operators often rely on various compromised machines to serve as their proxy hosts. Second, fast flux domains frequently change the compromised host they point to. Based on these observations, we can devise three types of features. First, IP diversity-based features rely on indicators such as the total number of IPs used, the number of different geolocation, the number of ISPs used, the number of ASNs and the type of IP address (mobile, residential, data center, etc.). Second, we also look at these features' entropy to capture how evenly they are distributed across these dimensions. Third, fast flux operators might quickly ramp up the number of IPs a domain resolves to when they start their campaigns. Thus,

we can protect against this approach by using features that calculate the change in the number of IPs used and their diversity.

Between December 2020 and January 2021, our classifier detected 2,679 fast flux domains, finding on average around 300 domains every week. We regularly identify hundreds of fast flux domains on the same day with similar name patterns.

Additionally, we protect users from DGA domains leveraging another detector. Traditional malicious domain detection and blocking methods either pre-build a denylist or cannot achieve realtime defense. Unfortunately, the disposable nature and virtually infinite size of DGA domains make these methods insufficient for DGA protection.

To stop new DGA domains when they are first seen, we developed a novel system that is able to detect DGA domains at resolution time. To achieve this, our firewalls intercept and inspect every DNS query. The DNS query is then simultaneously sent to DNS resolvers and our DGA detector in the cloud. Our detector inspects every single DNS query individually and returns a verdict to our firewalls before the corresponding DNS response is returned to users. If the queried domain is detected as DGA, our firewalls can block the DNS response or instead return a pre-specified DNS response like the IP addresses of DNS sinkholes. The core of our DGA detector is a machine learning (ML) model built upon a list of domain characteristics, such as the randomness of the root domain name (i.e., “foo” for “foo.com”). The output of the ML model not only contains a verdict denoting whether the domain is DGA, but also provides a malware family if the domain is indeed DGA. A full list of our DNS Security Analytics is also available.

Real-World Fast Flux Case Studies

Case Study: Social Engineering Campaign

We found an elaborate social engineering campaign hosted on `heygamersnort[.]at` leveraging fast fluxing to avoid detection. These cybercriminals lure victims by promising them a high payout in a short time period. These types of campaigns can be leveraged both for phishing and scamming users to pay cybercriminals.

The cybercriminals advertised their campaign in several ways. They sent out spam emails with the subject "`Earn 15.000 Euro every Month`" and the body containing URL `hXXp[:]//heygamersnort[.]at?gcGDRAewqASzXFDXcGCHjBJnhBGvFCCDRXTCyVBunINHBYGTFCRx` (spam email source). They also set up websites, as shown in Figure 5, with the same content as the spam email.

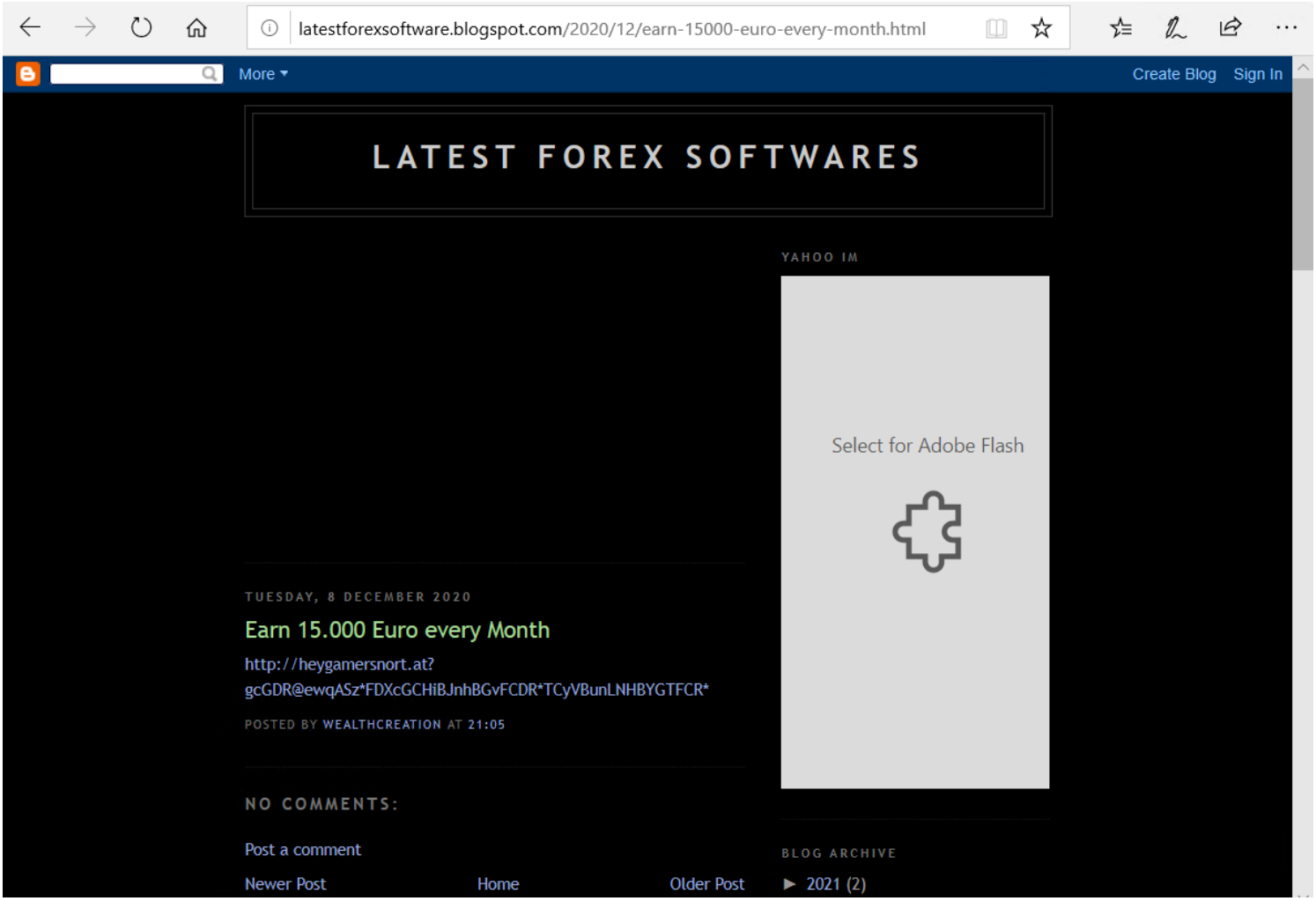

Figure 5. Screenshot of hXXp[:]//latestforexsoftware.blogspot[.]com/2020/12/earn-15000-euro-every-month.html.

What distinguishes this campaign from others is that the cybercriminals set up relatively high-quality websites in many different languages (including Dutch, French, German, Italian, Danish, Czech and English), as shown in Figure 6. We observed the domain `heygamersnort[.]at` resolving to over 200 IP addresses in less than two months, where these IP addresses are located all around the globe (mainly Eastern Europe, the Middle East and Central and South America) at different ISPs.

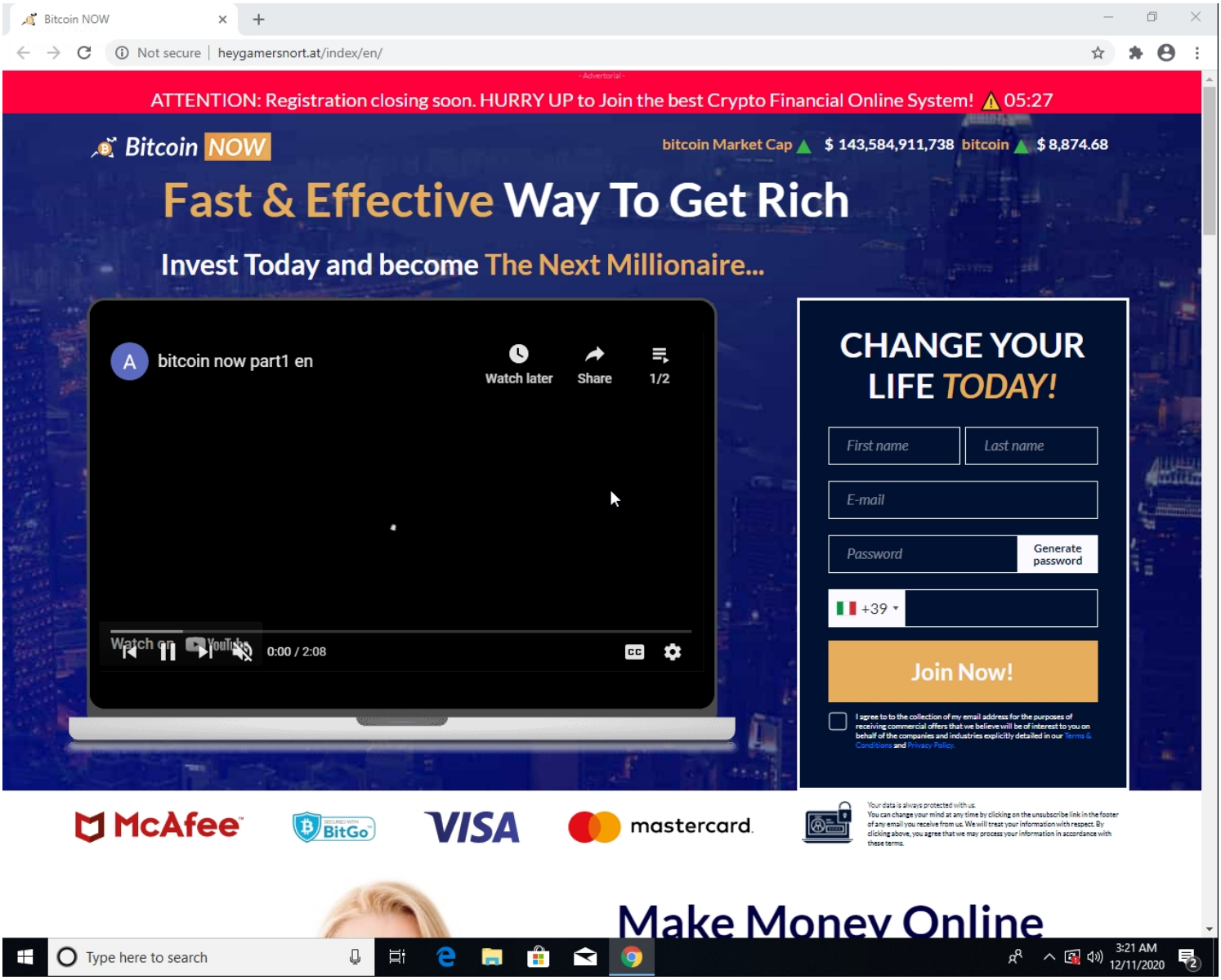

Figure 6.a. heygamersnort[.]at campaign in English (screenshot source urlscan.io and Joe Sandbox).

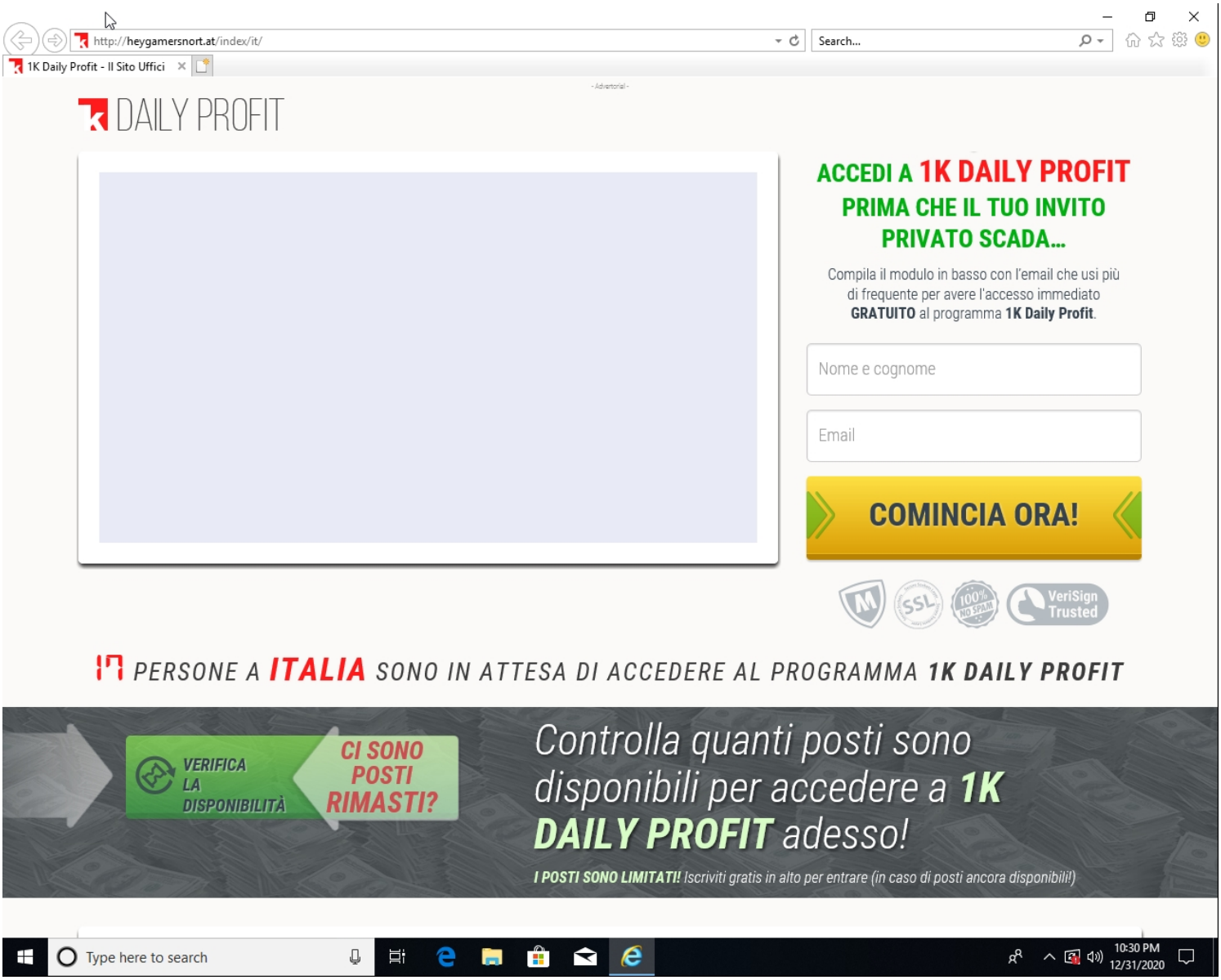

Figure 6.b. heygamersnort[.]at campaign in Italian (screenshot source urlscan.io and Joe Sandbox).

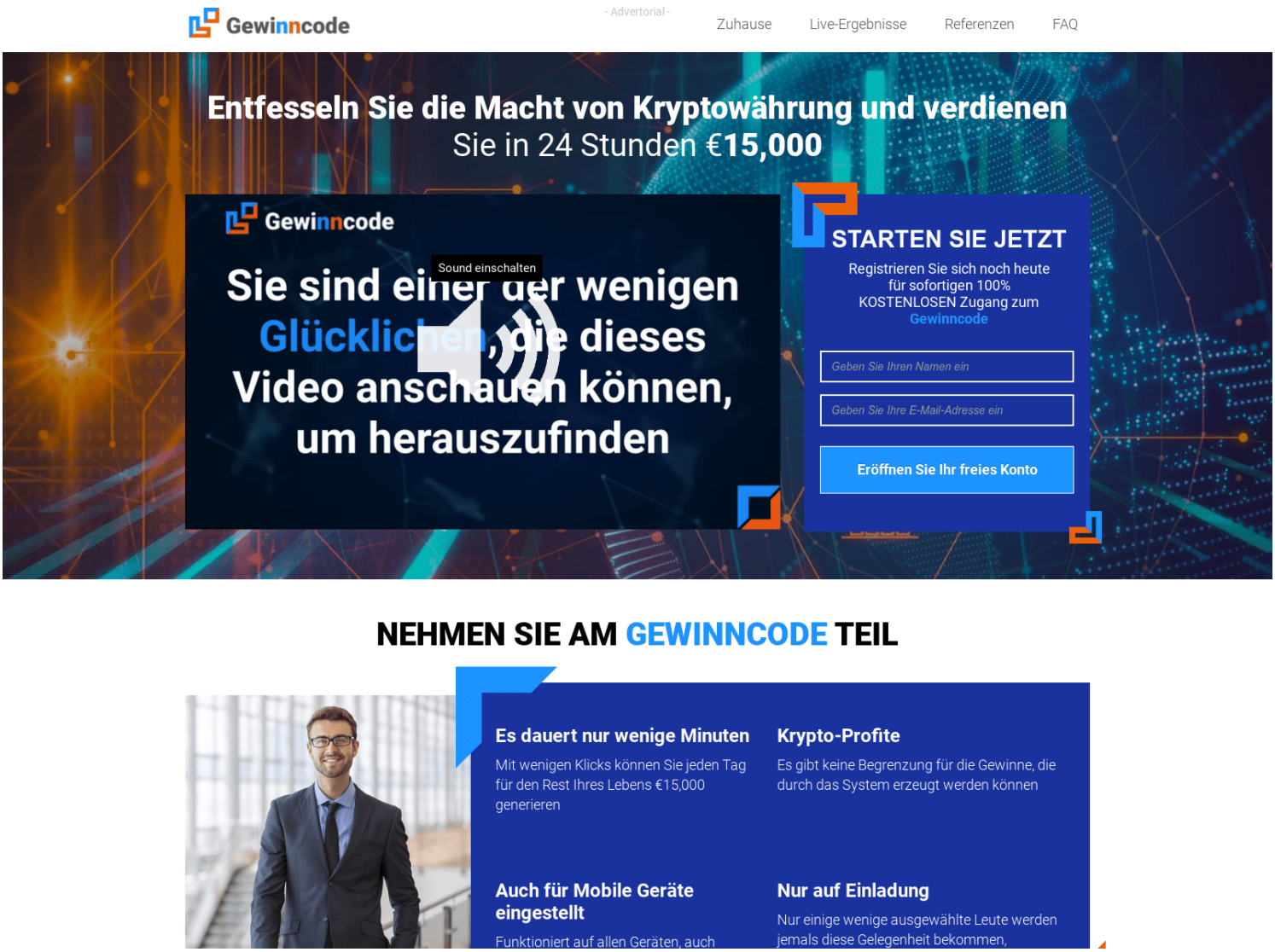

Figure 6.d. heygamersnort[.]at campaign in German (screenshot source urlscan.io and Joe Sandbox).

Figure 6.c. heygamersnort[.]at campaign in English (screenshot source urlscan.io and Joe Sandbox).

Users can look for multiple warning signs to avoid similar social engineering phishing and scam websites. First, when it looks “too good to be true,” then very likely, it is. If the cybercriminals could make 15,000 Euros a month easily, then they would not advertise how they do it. Second, these sites always convey a sense of urgency – they may say, “Only 28 free places left,” or include a timer counting down, making you think you only have a couple minutes to get rich.

Case Study: Smoke Loader Campaign C2

Our fast flux detector found multiple C2 domains related to the Smoke Loader malware family, such as `jamb2[.]monster`, `tinnys[.]monster` and `netvxi[.]com`. Unit 42 researchers have reported multiple instances of the Smoke Loader malware family, for example, related to a banking Trojan and a fake tsunami warning spam campaign.

Smoke Loader is a modular malware. When installed, it acts as a backdoor and allows attackers to download further malicious payloads from the C2 servers, which could be anything from ransomware to info stealers. The domains we discovered resolved to nearly 100 IP addresses in less than a two-week timeframe, where these IP addresses are located at many different ISPs and geolocations (mainly Eastern Europe, the Middle East and Central America).

Case Study: Illicit Gambling and Adult Sites

We found several large clusters of domains leveraging fast flux networks and at the same time hosting gambling and adult content. These sites are almost always in Chinese, as their activity is illegal in China. To evade detection and blocking, they usually use fast flux techniques. Typically, we observe that these domains point to hundreds of different IP addresses located all over the world (mainly North America, Western Europe and Asia). Furthermore, they only point to a given IP address a couple of times.

The domain `5651v[.]com` (shown in Figure 7) is an example of such a gambling site resolving many IP addresses. The domain `99guise[.]com` (Figure 8) is a typical example of gambling and adult content listing sites with dozens of links on the page.

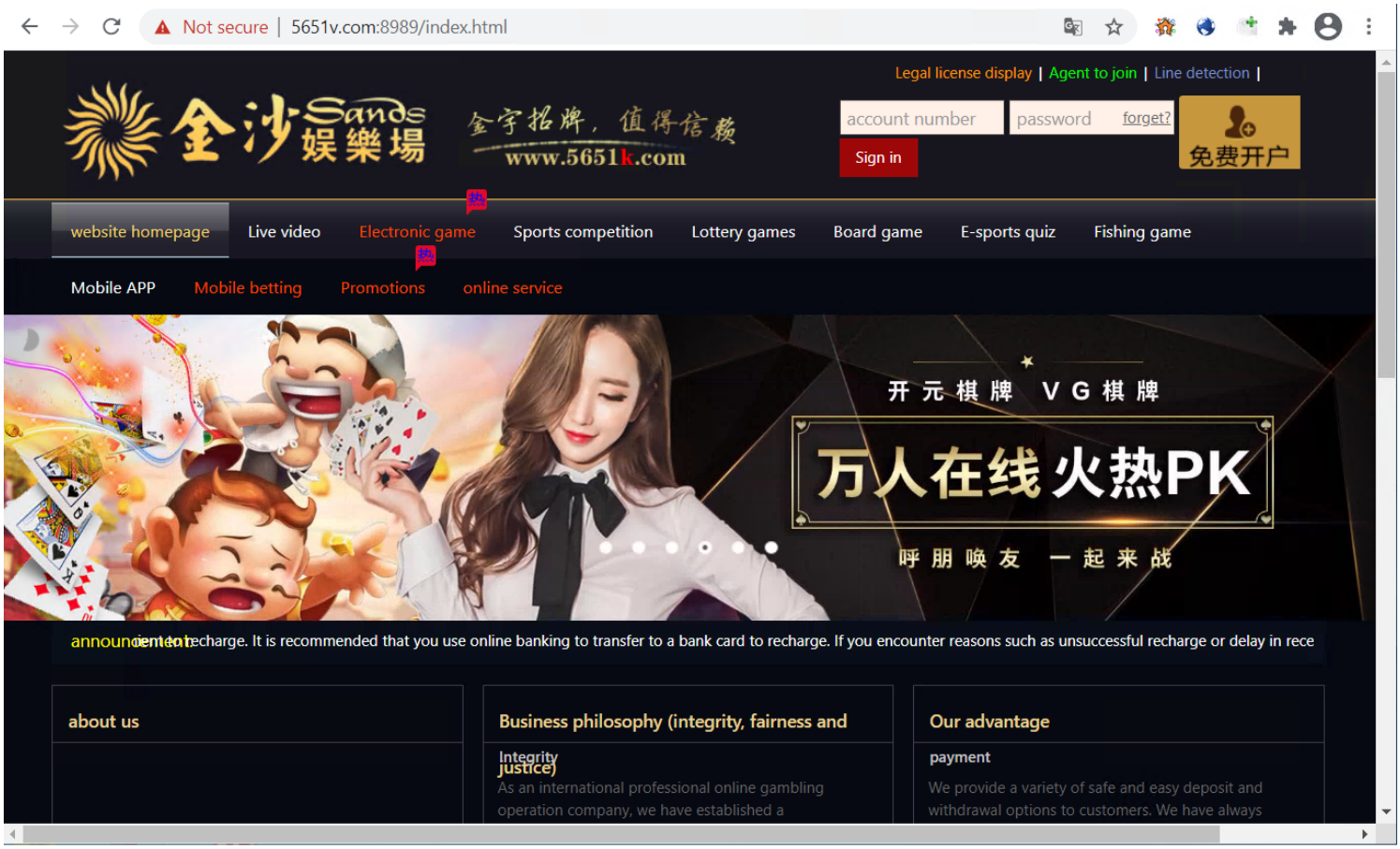

Figure 7. Example of a gambling page, 5651v[.]com (translated to English).

Figure 8. 99guise[.]com listing different gambling sites and adult content (not shown).

Worse than the illicit techniques they use to evade authorities, many of these sites offer mobile downloads to users that can frequently be malicious. The domains `612852[.]com` and `por99f9yw[.]com` (Figures 9 and 10) both offer file downloads that were found to be malicious.

Figure 9. 612852[.]com offering an app download for Android.

Figure 10. por99f9yw[.]com offering app download for Android and IOS.

Conclusion

There are various tactics and techniques available for cybercriminals to harden their systems against takedown and denylisting efforts such as fast fluxing, double fluxing and DGAs. While more basic techniques can be easily countered, advanced techniques result in a cat-and-mouse game between cybercriminals and law enforcement. Double fluxing can make IP-based denylists and host takedowns ineffective. DGA domains make static domain denylists and domain takeovers less effective.

At Palo Alto Networks, we automatically detect fast flux and DGA domains to protect our customers. Our detection results are built into multiple Palo Alto Networks Next-Generation Firewall security subscriptions, including URL Filtering and DNS Security.

Acknowledgements

We would like to thank Arun Kumar and Jun Javier Wang for their help with improving this blog post.

Indicators of Compromise (IOCs)

Social Engineering Campaign

- `heygamersnort[.]at`
- `hXXp[:]//heygamersnort[.]at?gcGDRAewqASzXFDXcGCHjBJnhBGvFCCDRXTCyVBunINHBYGTFCRx`
- `latestforexsoftware.blogspot[.]com`
- `hXXp[:]//latestforexsoftware.blogspot[.]com/2020/12/earn-15000-euro-every-month.html`

Smoke Loader C2 Campaign

- `Jamb2[.]monster`
- `Tinnys[.]monster`
- `Netvxi[.]com`

Illicit Adult and Gambling Sites

- `5651v[.]com`
- `99guise[.]com`
- `612852[.]com`
- `por99f9yw[.]com`

Additional Resources

- Toward Ending the Domain Wars: Early Detection of Malicious Stockpiled Domains — Unit 42, Palo Alto Networks