ipt_geofence: Protecting Networks using GeoFencing, Blocklists and Service Analysis

Luca Deri <deri@ntop.org>
@ntopng

In a Nutshell...

- Cybersecurity attacks are constantly increasing and this trend is (unfortunately) not going to stop any time soon.
- Linux/BSD Firewalls are able to efficiently handle IP/port-based policies but more complex tasks such as geofencing (e.g. allow SSH traffic to host X from country A, B, and C) is a nightmare to configure.
- How can I "anticipate" a problem? Perhaps blocklists area good idea...
- Ok but what about cyberattacks on (encrypted) services such as email or Wordpress?
- This has been the motivation to create a single (open source of course) tool to do all this in once.

Before ipt_geofence: Geofencing [1/2]

```
#!/bin/bash
# Purpose: Block all traffic from AFGHANISTAN (af) and CHINA (CN). Use ISO code. #
# See url for more info - http://www.cyberciti.biz/faq/?p=3402
# Author: nixCraft <www.cyberciti.biz> under GPL v.2.0+
# --------------------------------------------------------------------
ISO="XX" <--- Country to ban

### Set PATH ###
IPT=/sbin/iptables
WGET=/usr/bin/wget
EGREP=/bin/egrep

### No editing below ###
SPAMLIST="countrydrop"
ZONEROOT="/root/iptables"
DLROOT="http://www.ipdeny.com/ipblocks/data/countries"

cleanOldRules(){
    $IPT -F
    $IPT -X
    $IPT -t nat -F
    $IPT -t nat -X
    $IPT -t mangle -F
    $IPT -t mangle -X
    $IPT -P INPUT ACCEPT
    $IPT -P OUTPUT ACCEPT
    $IPT -P FORWARD ACCEPT
}

# create a dir
[ ! -d $ZONEROOT ] && /bin/mkdir -p $ZONEROOT

# clean old rules
cleanOldRules
```

```
# create a new iptables list
$IPT -N $SPAMLIST

for c  in $ISO
do
    # local zone file
    tDB=$ZONEROOT/$c.zone

    # get fresh zone file
    $WGET -O $tDB $DLROOT/$c.zone <--- Download zone file

    # country specific log message
    SPAMDROPMSG="$c Country Drop"

    # get
    BADIPS=$(egrep -v "^#|^$" $tDB)
    for ipblock in $BADIPS
    do
  # $IPT -A $SPAMLIST -s $ipblock -j LOG --log-prefix "$SPAMDROPMSG"
   $IPT -A $SPAMLIST -s $ipblock -j DROP
    done
done

# Drop everything
$IPT -I INPUT -j $SPAMLIST
$IPT -I OUTPUT -j $SPAMLIST
$IPT -I FORWARD -j $SPAMLIST

exit 0
```

Before ipt_geofence: Geofencing [2/2]

- Limitations
  - Dummy approach (download zone, load it in the firewall...)
  - Not too flexible (tricks can be applied) for things like:
    - Allow all countries to access by TCP/80 and TCP/443
    - Allow only my home country to access admin ports (e.g. TCP/22).
    - Allow ASN X (e.g. my mobile network provider) traffic to connect to my WireGuard VPN port and block all the rest.
    - Linux and BSD firewalls work differently, need to use two different solutions.

- Note: geofencing is not the ultimate solution to cybersecurity but it can help to mitigate risks and drop unwanted visitors on core services (e.g. SSH).

Blocklists [1/3]

- IP blacklists are widely used to increase network security by preventing (i.e. anticipate) communications with peers that have been marked as malicious.
- There are several commercial offerings as well as several free-of-charge blacklists maintained by volunteers on the web.
- Using blacklists to block IP addresses (hence the term "blocklist") is a good mechanism for preventing hosts with bad reputation to connect to our service.

Blocklists [2/3]

- What about blacklist effectiveness?

Evaluating IP Blacklists Effectiveness

Luca Deri, Francesco Fusco

IP blacklists are widely used to increase network security by preventing communications with peers that have been marked as malicious. There are several commercial offerings as well as several free-of-charge blacklists maintained by volunteers on the web. Despite their wide adoption, the effectiveness of the different IP blacklists in real-world scenarios is still not clear. In this paper, we conduct a large-scale network monitoring study which provides insightful findings regarding the effectiveness of blacklists. The results collected over several hundred thousand IP hosts belonging to three distinct large production networks highlight that blacklists are often tuned for precision, with the result that many malicious activities, such as scanning, are completely undetected. The proposed instrumentation approach to detect IP scanning and suspicious activities is implemented with home-grown and open-source software. Our tools enable the creation of blacklists without the security risks posed by the deployment of honeypots.

Subjects: **Cryptography and Security (cs.CR)**; Networking and Internet Architecture (cs.NI)
Cite as: arXiv:2308.08356 **[cs.CR]**
(or arXiv:2308.08356v1 **[cs.CR]** for this version)
https://doi.org/10.48550/arXiv.2308.08356

Submission history

From: Luca Deri [view email]
[v1] Wed, 16 Aug 2023 13:29:28 UTC (149 KB)

https://arxiv.org/pdf/2308.08356.pdf

Blocklists: Lessons Learnt [3/3]

- Fact: IP blacklists are "regional": 30% or the attackers do not roam across the Internet.
- Solution: We need local blacklists created in the region/ISP where our services live.

- Fact: In one day, 70% of the attackers attack other locations.
- Lesson Learnt: Blacklists need to be constantly updated to be effective.

- Fact: some blacklists are outdated, coarse (i.e. large CIDR), and with low match rate.
- Lesson Learnt: Use fresh high-quality blacklists to avoid false positives.

Tracking Attacks on Services [1/3]

- Beside blocklist usage, public services such as email and web cannot be restricted based on IP, ASN or protocol (they are public by definition).
- Most traffic is encrypted (e.g. HTTPS) or if cleartext it becomes encrypted (e.g. SMTP STARTTLS), making passive traffic analysis useless.

https://gist.github.com/jgriffyndor/4c70f9a4bc642f6573320711b7fb5a8d#file-smtp-starttls

SMTP StartTLS — Raw

```
1   $ telnet smtp.sendgrid.net 25
2   Trying 167.89.118.58...
3   Connected to smtp.sendgrid.net.
4   Escape character is '^]'.
5   220 SG ESMTP service ready at ismtpd0017p1las1.sendgrid.net
6   EHLO
7   250-smtp.sendgrid.net
8   250-8BITMIME
9   250-PIPELINING
10  250-SIZE 31457280
11  250-STARTTLS
12  250-AUTH PLAIN LOGIN
13  250 AUTH=PLAIN LOGIN
```

Tracking Attacks on Services [2/3]

- Service logs are the only way to detect when something went wrong at service level

access.log:167.71.204.176 - - [31/Jan/2024:03:17:25 +0100] "GET /?2bhUqOt4gXp952HAfQ9G4ggBTNQ=../../../../../../../../etc/passwd&2bhUqOt4gXp952HAfQ9G4ggBTNQ=1%20and%20updatexml(1,concat(0x7e,(select%20md5(97350))),1) HTTP/1.1" 200 440 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2866.71 Safari/537.36"

Feb 1 16:28:51 mail postfix/smtpd[3255487]: too **many** errors after AUTH from unknown[60.161.215.7]
Feb 1 16:29:32 mail postfix/smtpd[3255427]: too **many** errors after AUTH from unknown[59.46.193.187]
Feb 1 16:30:39 mail postfix/smtpd[3255392]: too **many** errors after AUTH from unknown[183.167.230.105]
Feb 1 16:39:37 mail postfix/smtpd[3256444]: too **many** errors after RCPT from noreply17.bncartes.com[138.197.173.107]
Feb 1 16:46:25 mail postfix/smtpd[3256519]: too **many** errors after RCPT from unknown[151.11.48.122]
Feb 1 16:54:23 mail postfix/smtpd[3256882]: too **many** errors after RCPT from unknown[151.11.48.122]

Tracking Attacks on Services [3/3]

There are various tools for monitoring applications logs:

- fail2ban
- SSHguard

- Various Wordpress log monitoring plugins.

- Opensource solutions exists already but they are
  - Limited to the host where the software is active.
  - They block IPs using the firewall adding/removing IPs.
  - Centralised monitoring or supervision needs to be created on top of the tools.

Welcome to ipt_geofence [1/2]

- Since tools for geofencing, blocklists, service monitoring.
- Written in C++ and sitting on top of Linux/BSD.
- Single configuration file for defining policies.
- Early extensible (e.g. create your own "watcher") for your custom services.
- It automatically blocks/unblocks, refreshes blocklists without any scripting or cron-based tools.
- Results can be shared via ZMQ, Telegram or custom shell commands.

Configuration [1/2]

```
{
    "queue_id": 0,
    "markers": {
        "pass": 1000,
        "drop": 2000
    },
    "default_policy": "DROP",
    "monitored_ports": {
        "tcp": [22, 80, 443],
        "udp": [],
        "ignored_ports": [123],
        "honeypot_ports": ["51000-56000","50000-56100", "51000-52000",10,20,30]
    },
    "policy": {
        "drop": {
            "countries_whitelist": ["IT", "DE", "CH", "NL"],
            "continents_whitelist": ["NA"]
        },
        "pass": {
            "countries_blacklist": ["RU", "BY"],
            "continents_blacklist": []
        }
    },
```

Configuration [2/2]

```
    "blacklists": [
        "https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/dshield_7d.netset",
        "https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/alienvault_reputation.ipset",
        "https://feodotracker.abuse.ch/downloads/ipblocklist_recommended.txt",
        "https://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt",
        "https://feodotracker.abuse.ch/downloads/ipblocklist.txt",
        "https://sslbl.abuse.ch/blacklist/sslipblacklist.txt"
    ],
    "watches": [
        { "name" : "mail", "cmd": "/usr/share/ipt_geofence/scripts/mail.py" },
        { "name" : "auth", "cmd": "/usr/share/ipt_geofence/scripts/auth.py" }
    ],
    "telegram": {
        "bot_token": "",
        "chat_id": ""
    },
    "cmd": {
        "ban": "",
        "unban": ""
    },
    "zmq": {
        "url": "",
        "encryption_key": ""
    }
}
```

ipt_geofence Internals

- Written in C++ (engine) and Python (watchers).
- Traffic is intercepted with NF_QUEUE (Linux) and NETMAP (BSD).
- Based on the configuration a decision is made an a marker is set (Linux) or continuously dropped (BSD).
- (Linux) Only the first packet of a flow is observed with no continuous traffic analysis necessary thanks to markers.
- File "watchers" are spawn and used to complement network-based alerting with services.
- Geolocation is based on MaxMind or IPDB files.
- Blacklists are automatically downloaded, refreshed, and reloaded every night.

Alert Notifications: Telegram

What's Next?

- ipt_geofence protects ntop servers since 3 years and it has been useful to keep services up and reduce workload by stopping scanners/attackers.
- We would like to combine this tool to dynamically create blocklists, share results in realtime (no daily blacklist download), and make the Internet more secure for everyone.
- We are making experiments with a work in progress "cloud" to share results with a central location that can analyse them and redistribute to future ipt_geofence versions or share them with other applications in an anonymous format (no more lengthly text-based blocklists).

https://github.com/ntop/ipt_geofence