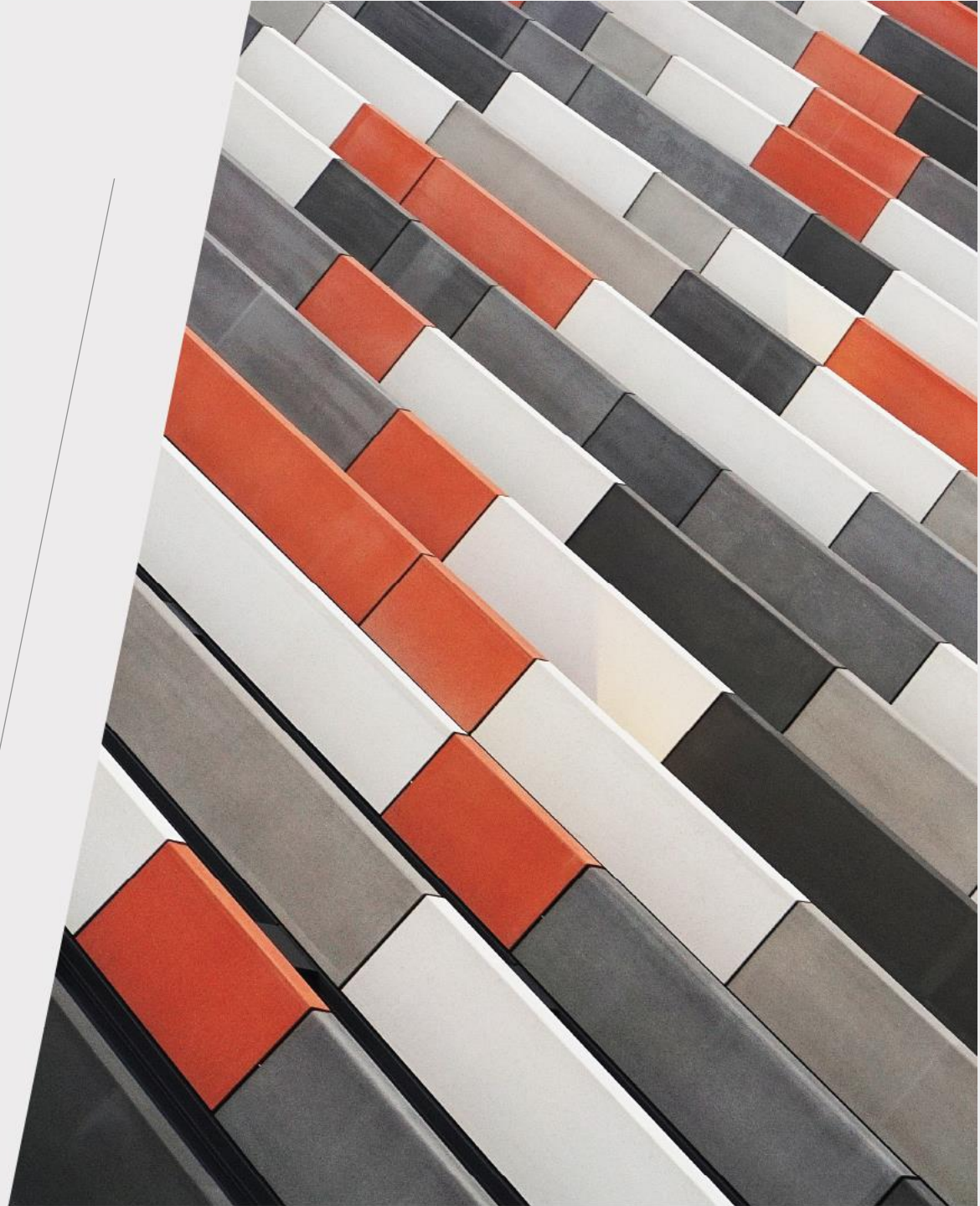
TWELVESEC

Your Partner in Cyber Security

Creating a Resilient Red Team Infrastructure



www.twelvesec.com
hello@twelvesec.com

Content

1. **Intro**
2. **Presentation Expectations**
3. **Why is this needed**
    - What does a Red Teamer use during an engagement
    - How would the infrastructure look like
    - “Traditional” way to build the infrastructure
4, **The need of IaaC**
    - Problems for the “Traditional” way
    - How to fix them using IaaC
5. **Project Overview & Customization**
    - File Structure & Usage
    - Dashboards overview
    - Customization & IOC
    - Costs
6. **Automation Leftovers**
7. **Extra**
8. **Demo**

`whoami`

- Senior Penetration Tester eJPT, PNPT, OSCP, OSEP, CRTO, CRTL certified
- In love with Red Teaming: Phishing, AD exploitation and Evading Defenses
- Poker Fanatic
- Music & Hi-Fi Systems addict

What is this about ?

- How to build a resilient red team infrastructure
- What resources are necessary to accomplish that considering modern state of cybersecurity protections
- How to protect your red team infrastructure
- How feasible is this approach from a financial perspective
- What aspects are yet to be manually required

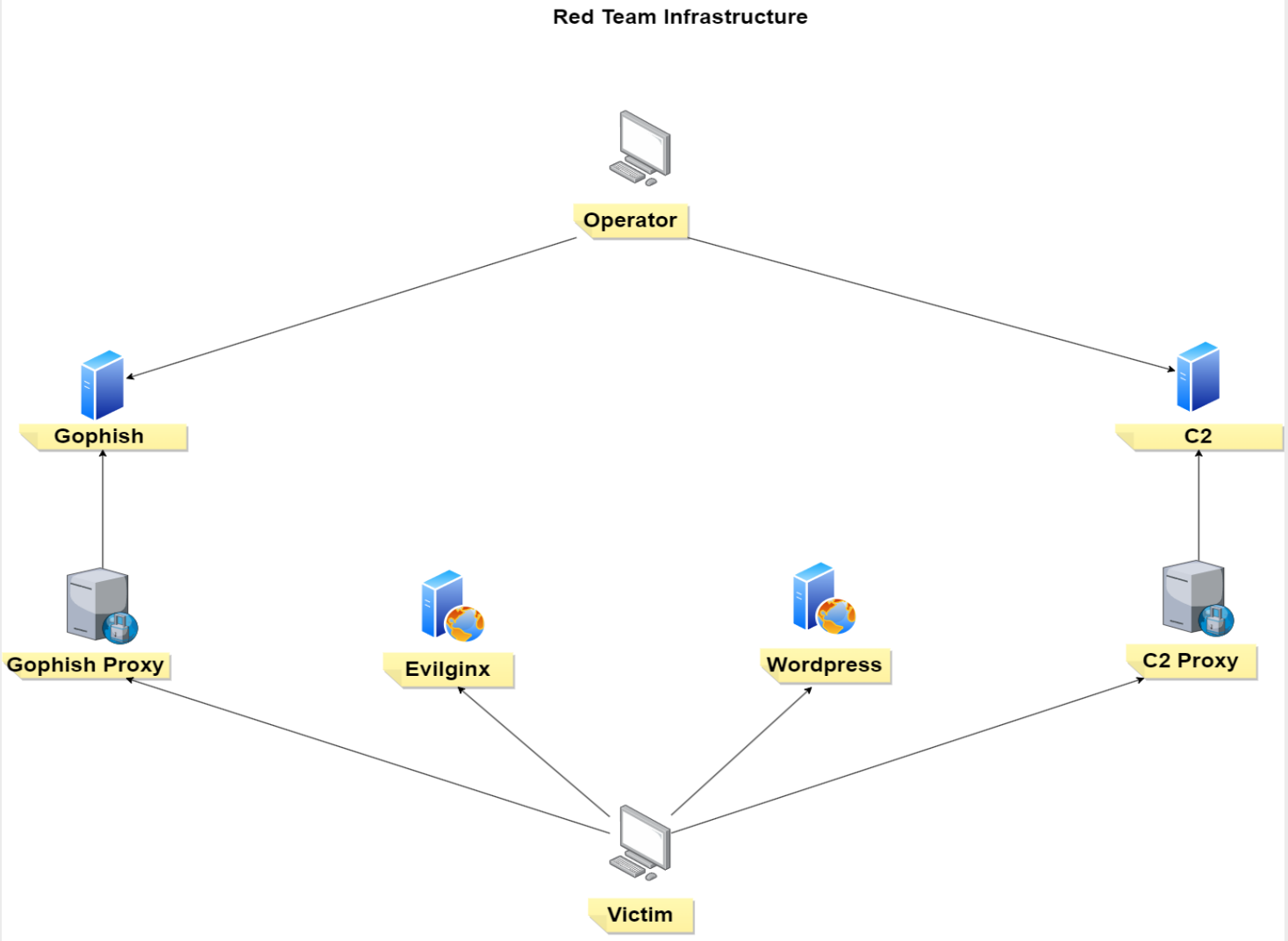
What is this not about ?

- Deep dive into Terraform coding principles
- Line by line code analysis
- State of the art ideas & principles
- Bullet proof red team infrastructures
- Open source project (yet)

What does a RT requires ?

- a host with public IP to deploy a C2
  - ✓ Metasploit/Sliver (open-source) or CobaltStrike/BruteRatel (paid)
- a host with public IP to deploy a phishing framework
  - ✓ Gophish
- A host with a public IP to store/manage phishing templates
  - ✓ Evilginx
- multiple domains + custom DNS entries
  - ✓ GoDaddy/Namecheap
- multiple redirectors (HTTP, SMTP, DNS, SMB)
  - ✓ Socat/Lambda/SSH tunneling

Infrastructure Diagram



Requirements

- on premise servers + management (e.g. ESXi)

OR

- cloud provider (e.g. DigitalOcean)

How

- manually install each tool and set configurations options each time (e.g. firewall, Apache config)

OR

- do it manually once and then bundle the result (base image/packer) to simply reuse it

Problems

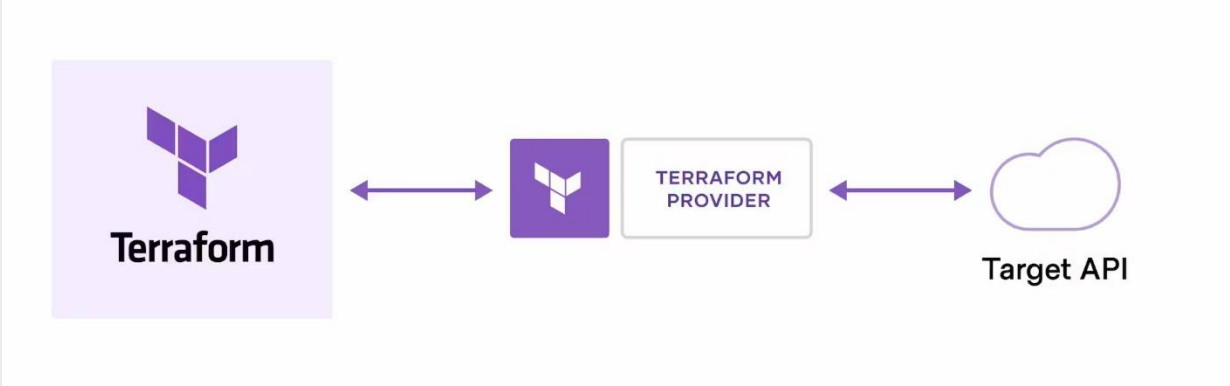
- You still need auxiliary scripts to set up images for each new engagement (set different whitelists, assign domain and subdomains)
- How do you hide your license keys/tokens if you want to automate installation through a script ?
- How much space do you need to store so many different bundles?
- What do you do when you want to replace and old tool with a new one ? (you have re-create the bundle)

Solution ?

We need to:

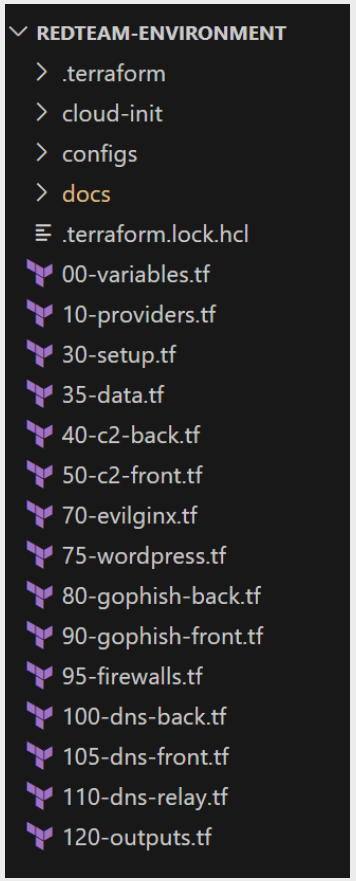
- Spawn and destroy hosts with a simple command
- Configure each host via code that can be easily modified/updated
- Import secrets/tokens during installation/configuration in a secure way

Result = Infrastructure as a Code => Terraform

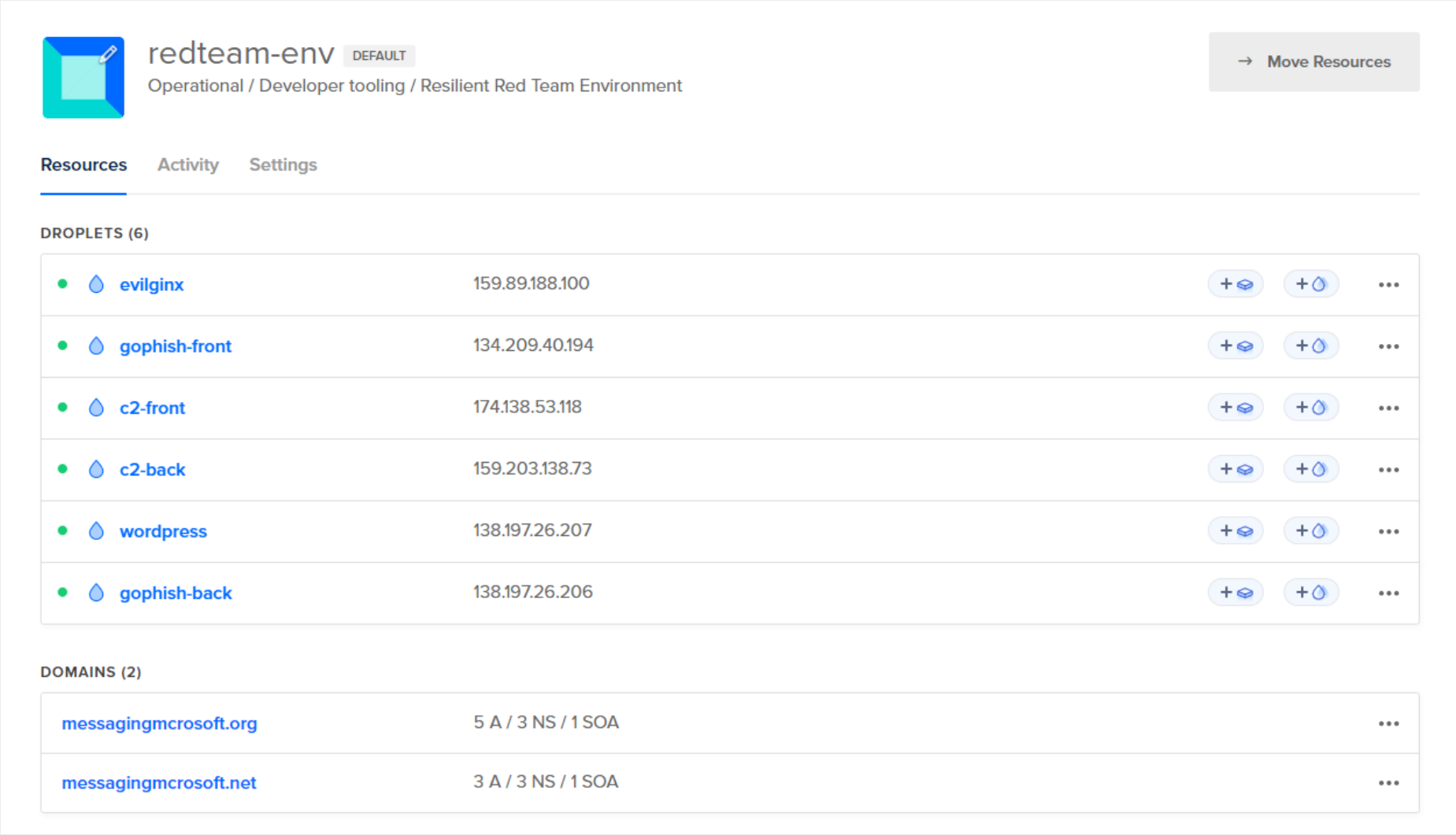


Solution ?

- cloud-init: YAML configuration file for each droplet on their first boot
- configs: Contains custom software configuration files
- variables: Global variables definition
- providers: Provider definition (Name, API Key/Token)
- setup: Define organization and workspace to be used for the project
- data: Defines data sources (local files, templates, DigitalOcean resources)
- c2-back: Configuration for the C2 server
- c2-front: Configuration for the C2 redirector
- evilginx: Configuration for the evilginx server
- wordpress: Configuration for the WordPress server
- gophish-back: Configuration for the Gophish server
- gophish-front: Configuration of the Gophish redirector
- firewalls: Define firewall rules (inbound and outbound) for the created droplets
- dns-main, dns-redirect, dns-relay: DNS records to be created for the acquired domains
- outputs: Verbose output to be generated at the end of a successful compilation



Solution ?



Solution ?

← Domains

messagingmcrosoft.org

in redteam-env

Create new record

A AAAA CNAME MX TXT NS SRV CAA

Use @ to create the record at the root of the domain or enter a hostname to create it elsewhere. A records are for IPv4 addresses only and tell a request where your domain should direct to.

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
Enter @ or hostname *	Select resource or enter custom IP	Enter TTL 3600 ✓	Create Record

DNS records

Type	Hostname	Value	TTL (seconds)	
A	dev.messagingmcrosoft.org	directs to 174.138.53.118	120	More
A	o365.messagingmcrosoft.org	directs to 134.209.40.194	60	More
A	messagingmcrosoft.org	directs to 159.89.188.100	60	More
A	*.messagingmcrosoft.org	directs to 159.89.188.100	60	More
A	www.messagingmcrosoft.org	directs to 138.197.26.207	60	More
NS	messagingmcrosoft.org	directs to ns1.digitalocean.com.	1800	More
NS	messagingmcrosoft.org	directs to ns2.digitalocean.com.	1800	More
NS	messagingmcrosoft.org	directs to ns3.digitalocean.com.	1800	More

Solution ?

Networking

Domains | Reserved IPs | Load Balancers | VPC | **Firewalls** | PTR records

Create Firewall

Name	Droplets	Rules	Created	
http-front	4	4	15 minutes ago	More
dns-ssh-droplets	6	2	15 minutes ago	More
http-c2-back	1	2	15 minutes ago	More
http-gophish-back	1	1	15 minutes ago	More
mysql-evilginx	1	1	15 minutes ago	More
mysql-gophish-back	1	1	15 minutes ago	More
http-back	2	2	15 minutes ago	More
login-gophish-back	1	1	15 minutes ago	More
smtp-gophish-back	1	1	15 minutes ago	More

Customizations ?

- SSL Everywhere
  - ✓ self-signed certificates for back hosts (gophish, C2 server)
  - ✓ let’s encrypt certificates (certbot) front facing hosts
- Remove IoC from Gophish and Evilginx
- Security through ~~obscurity~~ blacklisting

SSL Everywhere

```
write_files:
  - content: |
      SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
      SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
      SSLHonorCipherOrder On

      Header always set X-Frame-Options DENY
      Header always set X-Content-Type-Options nosniff

      SSLCompression off
      SSLUseStapling on
      SSLStaplingCache "shmcb:logs/stapling-cache(150000)"

      SSLSessionTickets Off
    path: /etc/apache2/conf-available/ssl-params.conf
    permissions: '0644'
    defer: true

  - content: |
      <IfModule mod_ssl.c>
        <VirtualHost _default_:443>
            ServerName ${front-domain}
            DocumentRoot /var/www/html
            ErrorLog $${APACHE_LOG_DIR}/error.log
            CustomLog $${APACHE_LOG_DIR}/access.log combined

            SSLEngine on
            SSLCertificateFile      /etc/letsencrypt/live/${front-domain}/cert.pem
            SSLCertificateKeyFile   /etc/letsencrypt/live/${front-domain}/privkey.pem

            RewriteEngine On
            RewriteRule ^.*$ http://${gophish-server}%%{REQUEST_URI} [P]
        </VirtualHost>
      </IfModule>
    path: /etc/apache2/sites-available/default-ssl.conf
    permissions: '0644'
    defer: true
```

```
  - content: |
      DefaultRuntimeDir $${APACHE_RUN_DIR}
      PidFile $${APACHE_PID_FILE}
      Timeout 300
      KeepAlive On
      MaxKeepAliveRequests 100
      KeepAliveTimeout 5
      User $${APACHE_RUN_USER}
      Group $${APACHE_RUN_GROUP}
      HostnameLookups Off
      ErrorLog $${APACHE_LOG_DIR}/error.log
      LogLevel warn
      IncludeOptional mods-enabled/*.load
      IncludeOptional mods-enabled/*.conf
      Include ports.conf
      <Directory />
        Options FollowSymLinks
        AllowOverride None
        Require all denied
      </Directory>
      <Directory /usr/share>
        AllowOverride None
        Require all granted
      </Directory>
      <Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
      </Directory>
      AccessFileName .htaccess
      <FilesMatch "^\.ht">
        Require all denied
      </FilesMatch>
      LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%%{Referer}i\" \"%%{User-Agent}i\"" vhost_combined
      LogFormat "%h %l %u %t \"%r\" %>s %O \"%%{Referer}i\" \"%%{User-Agent}i\"" combined
      LogFormat "%h %l %u %t \"%r\" %>s %O" common
      LogFormat "%%{Referer}i -> %U" referer
      LogFormat "%%{User-agent}i" agent
      IncludeOptional conf-enabled/*.conf
      IncludeOptional sites-enabled/*.conf
    path: /etc/apache2/apache2.conf
    permissions: '0644'
    defer: true
```

SSL Everywhere

```
- sed -i '5d' /etc/apache2/ports.conf
- service apache2 stop
- certbot certonly --standalone -d ${front-domain} --register-unsafely-without-email --agree-tos
- service apache2 start
- a2enmod ssl
- a2enmod headers
- a2enconf ssl-params
- a2ensite default-ssl
- a2enmod rewrite proxy proxy_http
- systemctl restart apache2
- reboot
```

```
- sed -i '5d' /etc/apache2/ports.conf
- openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt -subj "/C=US/ST=California/L=Los A
- a2enmod ssl
- a2enmod headers
- a2enconf ssl-params
- a2ensite default-ssl
```

Gophish IoCs ?

- Modify default 404.html page
  - ✓ Default page hash = gophish

- Modify default controllers/phish.go
  - ✓ Overwrite net.https Error with a custom one to set our own headers
  - ✓ Re-write gophish internal to allow templating of custom 404 pages

- Remove any strings associated with Gophish

```
sed -i 's/X-Gophish-Contact/X-Contact/g’          models/email_request_test.go
sed -i 's/X-Gophish-Contact/X-Contact/g' models/maillog.go
sed -i 's/X-Gophish-Contact/X-Contact/g' models/maillog_test.go
sed -i 's/X-Gophish-Contact/X-Contact/g' models/email_request.go
sed -i 's/X-Gophish-Signature/X-Signature/g' webhook/webhook.go
sed -i 's/const ServerName = "gophish"/const ServerName = "IGNORE"
/' config/config.go
sed -i 's/const RecipientParameter = "rid"/const RecipientParameter = "mailer"/g’ models/campaign.go
```

Or simply use https://github.com/puzzlepeaches/sneaky_gophish

Evilginx2 IoCs ?

7 evilginx/core/http_proxy.go

```diff
@@ -183,7 +183,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
183 183                                 pl_name = pl.Name
184 184                         }
185 185
186     -                   egg2 := req.Host
187 186                     ps.PhishDomain = phishDomain
188 187                     req_ok := false
189 188                     // handle session
@@ -350,7 +349,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
350 349                         }
351 350                 }
352 351
353     -               hg := []byte{0x94, 0xE1, 0x89, 0xBA, 0xA5, 0xA0, 0xAB, 0xA5, 0xA2, 0xB4}
354 352                 // redirect to login page if triggered lure path
355 353                 if pl != nil {
356 354                         _, err := p.cfg.GetLureByPath(pl_name, req_path)
@@ -383,9 +381,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
383 381                         req.Header.Del("Cookie")
384 382                 }
385 383
386     -               for n, b := range hg {
387     -                       hg[n] = b ^ 0xCC
388     -               }
389 384                 // replace "Host" header
390 385                 e_host := req.Host
391 386                 if r_host, ok := p.replaceHostWithOriginal(req.Host); ok {
@@ -398,8 +393,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
398 393                 // fix referer
399 394                 p.replaceHeaderWithOriginal(req, "Referer")
400 395
401     -               req.Header.Set(string(hg), egg2)
402     -
403 396                 // patch GET query params with original domains
404 397                 if pl != nil {
405 398                         qs := req.URL.Query()
```

Evilginx2 IoCs ?

14 evilginx2/core/http_proxy.go

```diff
@@ -565,11 +565,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
565 565                                                req.Body = ioutil.NopCloser(bytes.NewBuffer([]byte(body)))
566 566                                        }
567 567                                }
568     -                               e := []byte{208, 165, 205, 254, 225, 228, 239, 225, 230, 240}
569     -                               for n, b := range e {
570     -                                       e[n] = b ^ 0x88
571     -                               }
572     -                               req.Header.Set(string(e), e_host)
573 568
574 569                                 if pl != nil && len(pl.authUrls) > 0 && ps.SessionId != "" {
575 570                                         s, ok := p.sessions[ps.SessionId]
@@ -583,7 +578,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
583 578                                         }
584 579                                 }
585 580                         }
586     -                       p.cantFindMe(req, e_host)
587 581                 }
588 582
589 583
@@ -1545,14 +1539,6 @@ func (p *HttpProxy) getSessionIdByIP(ip_addr string) (string, bool) {
1545 1539         return sid, ok
1546 1540 }
1547 1541
1548     - func (p *HttpProxy) cantFindMe(req *http.Request, nothing_to_see_here string) {
1549     -         var b []byte = []byte("\x1dh\x003,)\",+=")
1550     -         for n, c := range b {
1551     -                 b[n] = c ^ 0x45
1552     -         }
1553     -         req.Header.Set(string(b), nothing_to_see_here)
1554     - }
1555     -
1556 1542 func (p *HttpProxy) setProxy(enabled bool, ptype string, address string, port int, username string, password string) error {
1557 1543         if enabled {
1558 1544                 ptypes := []string{"http", "https", "socks5", "socks5h"}
```

Evilginx3 IoCs ?

7 evilginx3/core/http_proxy.go

```diff
@@ -176,7 +176,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
176 176                 }
177 177
178 178                 req_url := req.URL.Scheme + "://" + req.Host + req.URL.Path
179     -               o_host := req.Host
180 179                 lure_url := req_url
181 180                 req_path := req.URL.Path
182 181                 if req.URL.RawQuery != "" {
@@ -327,7 +326,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
327 326                                 return p.blockRequest(req)
328 327                         }
329 328                 }
330     -               req.Header.Set(p.getHomeDir(), o_host)
331 329
332 330                 if ps.SessionId != "" {
333 331                         if s, ok := p.sessions[ps.SessionId]; ok {
@@ -509,7 +507,6 @@ func NewHttpProxy(hostname string, port int, cfg *Config, crt_db *CertDb, db *da
509 507
510 508                 // check for creds in request body
511 509                 if pl != nil && ps.SessionId != "" {
512     -               req.Header.Set(p.getHomeDir(), o_host)
513 510                 body, err := ioutil.ReadAll(req.Body)
514 511                 if err == nil {
515 512                         req.Body = ioutil.NopCloser(bytes.NewBuffer([]byte(body)))
@@ -1492,10 +1489,6 @@ func (p *HttpProxy) getPhishDomain(hostname string) (string, bool) {
1492 1489         return "", false
1493 1490 }
1494 1491
1495     - func (p *HttpProxy) getHomeDir() string {
1496     -         return strings.Replace(HOME_DIR, ".e", "X-E", 1)
1497     - }
1498     -
1499 1492 func (p *HttpProxy) getPhishSub(hostname string) (string, bool) {
1500 1493         for site, pl := range p.cfg.phishlets {
1501 1494                 if p.cfg.IsSiteEnabled(site) {
```

Blacklisting the Internet ?

- [**CONFIDENTIAL**] Both Google and Microsoft scan the entire public range to discover IPs, domain and services
  - ✓ easily detect phishing websites or suspicious/malicious services (e.g CobaltStrike server fingerprint, Metasploit listeners)
  - ✓ collect and store data to be able to classify it later (e.g. domain reputation)

BUT …

- They have known public IP ranges for this scanners/protections

BUT …

- [**CONFIDENTIAL**] Others do the same as Google and Microsoft (e.g. Threat Intelligence companies like Censys/Shodan)

BUT …

- We can configure strict firewall rules based on whitelist (block anything else).
- We can spin up Evilginx and log every IP trying to access it and store it in a file. Then, create the actual phishing page and blacklist every captured IPs from before.

Blacklisting the Internet ?

```
root@evilginx:/opt/evilginxbackup# ./build/evilginx -p phishlets/

                                  - --  Community Edition  -- -

                                by Kuba Gretzky (@mrgretzky)     version 3.0.0

[14:59:28] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-mastery (learn how to create phishlets)
[14:59:28] [inf] loading phishlets from: phishlets/
[14:59:28] [inf] loading configuration from: /root/.evilginx
[14:59:29] [inf] blacklist: loaded 3809 ip addresses and 9130 ip masks
[14:59:29] [inf] obtaining and setting up 1 TLS certificates - please wait up to 60 seconds...
[14:59:29] [inf] successfully set up all TLS certificates
```

Costs ?

- Terraform
  - ✓ free for local deployment
  - ✓ free for up to 500 resources/month for cloud deployment
- DigitalOcean
  - ✓ current project uses 6 droplets, each with a cost of 6$/month (c2-back, c2-front,evilginx,wordpress,gophish-front,gophish-back)

What’s left ?

- Find the right domains to purchase (API already existing)
- Change nameservers to DigitalOcean (mandatory requirement)
- Create/Configure SMTP relay
- Build domain reputation
- Link Gophish with Evilginx database to keep track of captured credentials

SMTP Relaying ?

Initially, the Gophish front droplet was installing and configuring it’s own SMTP relay to be used with the purchased domain. However, since last year, cloud providers have stopped supporting any SMTP traffic as it was widely abused by attackers for phishing scams. Therefore, we are left with the following options:

- Migrate the infrastructure from cloud to local provisioning with ansible
  - ✓ Lack of infrastructure lifecycle
  - ✓ Limited Windows support
  - ✓ Limited Cloud providers support
  - ✓ Does not scale as much as Terraform does

- SMTP relay using providers such as Microsoft/Google + business plan
  - ✓ It might be easier for them to detect your malicious activities as they have full control over your emails

- SMTP relay + domain authentication using providers such as SendGrid
  - ✓ Easy to use and Free up to 100 emails/day (20-90 dollars/month for up to 200k emails/day)

SMTP Relaying ?

Integrate using our Web API or SMTP Relay

- Overview
- 2 Integrate
- 3 Verify

How to send email using the SMTP Relay

1 **Create an API key**

This allows your application to authenticate to our API and send mail. You can enable or disable additional permissions on the API keys page.

"test_key" was successfully created and added to the next step.

SG.[redacted]

2 **Configure your application**

Configure your application with the settings below.

Server	smtp.sendgrid.net
Ports	25, 587 (for unencrypted/TLS connections) 465 (for SSL connections)
Username	apikey
Password	[redacted]

Domain Reputation ?

[**CONFIDENTIAL**] When sending emails from a custom domain, one aspect that influences if the email will be classified as malicious/spam/phishing is its reputation.

But how can you get a good reputation ?

- Buy a domain with a good reputation (silly but it works)
  - ✓ Monitor domain that are close to expiration
  - ✓ Find expired domains using https://www.expireddomains.net/

- Build reputation on your own
  - ✓ Create a landing website/blog using a CMS such as Wordpress
  - ✓ Populate it with relevant data based on your desired category (health, banking, finance are the most used ones by attackers) – ChatGPT might help with proper content
  - ✓ If you have time, use social media to promote your domain, send relevant emails, create blog posts about popular topics on the chosen category/field
  - ✓ “Warm-up” your mailbox

Domain Reputation ?

How to build a decent reputation when you are short on time ?

Manually issue categorization requests to vendors to evade proxy categorization/filtering

- https://sitereview.bluecoat.com/#/
- https://urlfiltering.paloaltonetworks.com/
- https://support.sophos.com/support/s/filesubmission?language=en_US
- https://global.sitesafety.trendmicro.com/feedback.php
- https://www.brightcloud.com/tools/url-ip-lookup.php
- http://csi.websense.com/
- https://archive.lightspeedsystems.com/
- https://sitelookup.mcafee.com/

So wait, is there any tool that would automate domain categorization requests?

YES

https://github.com/mdsecactivebreach/Chameleon

BUT

- This tool has not been updated in 3 years
- None of the vendors behave the same as they did before + they all have some type of captcha

Domain Reputation ?

The idea was there, so I just re-implemented everything

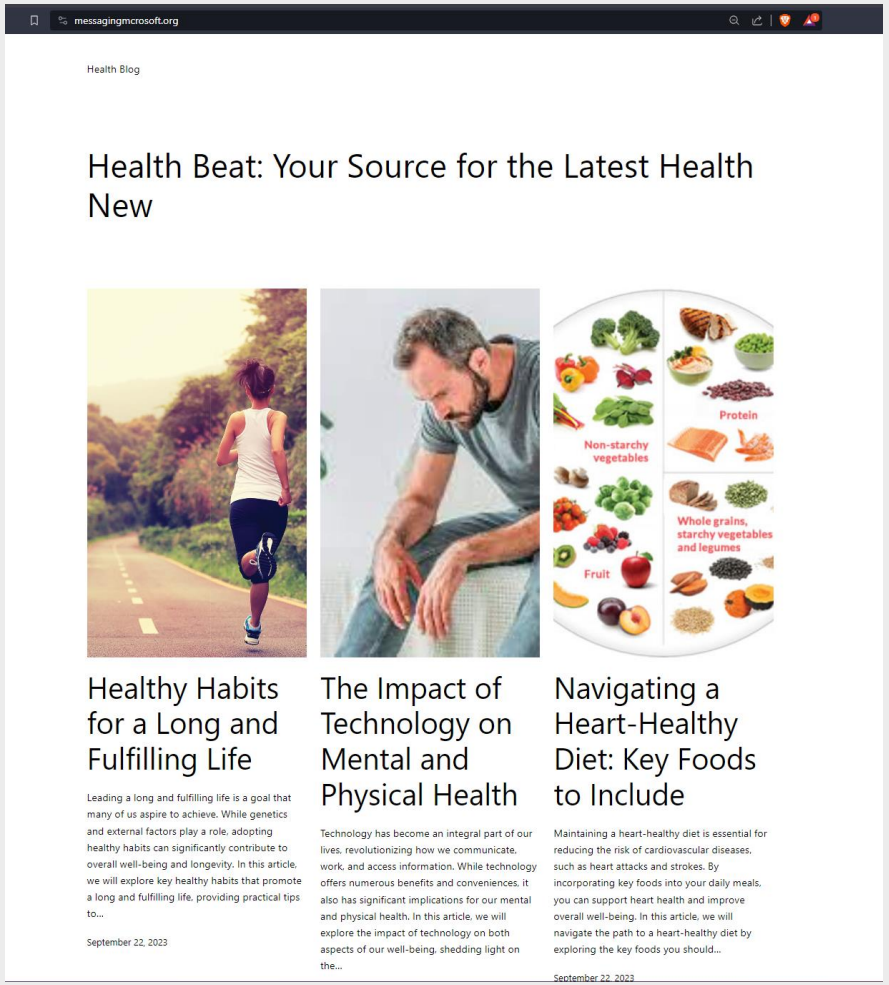
- Do everything with Selenium
- Use Mullvad VPN to switch IP to avoid getting blocked
- Implemented Captcha Solver using Ffmpeg

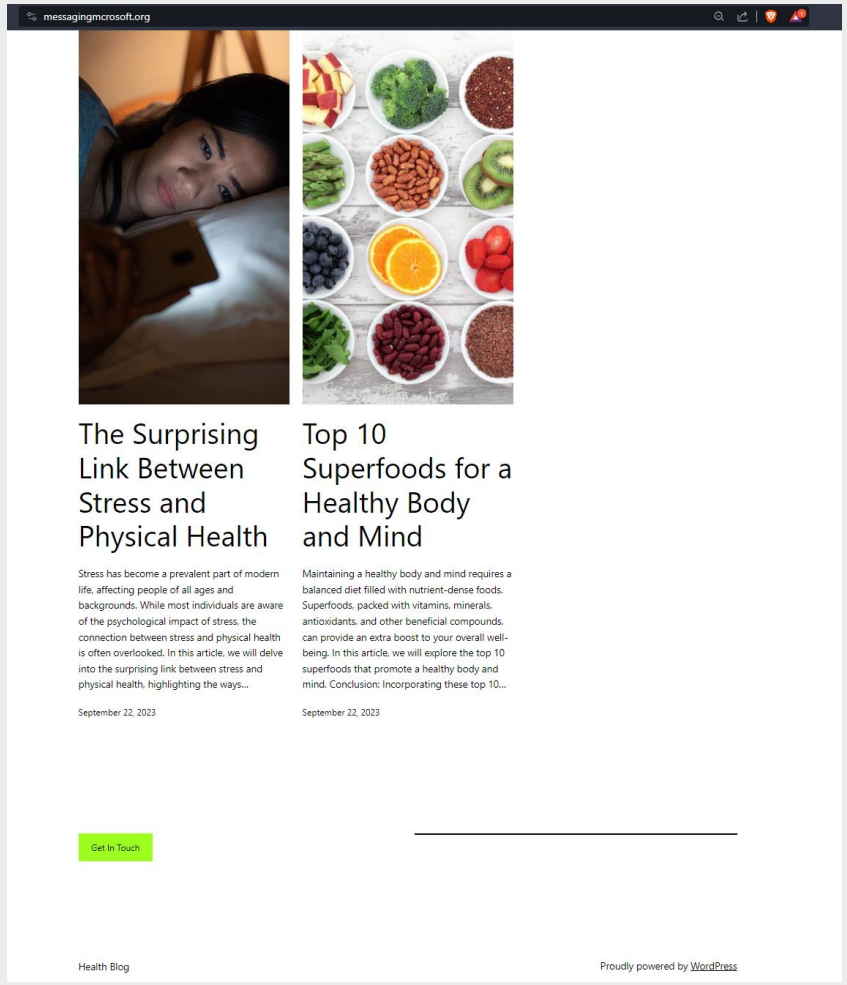
Vendor response?

- Talos Intelligence, Bright Cloud, Palo Alto simply classified it as requested (some approved via email, some just updated their records)
- Some are still marked as “Newly Observed”
- Some were classified simply as “IT”

```
PS H:\Projects\CheckDomains\chameleonv2> python3 main.py --proxy a --check --domain [redacted]
2023-09-06 04:15:05,954 - INFO - ----------------------------------------------------------------

DevTools listening on ws://127.0.0.1:8128/devtools/browser/955ee87b-5d78-4621-a7b3-68a2e5f25d63
2023-09-06 04:15:18,171 - INFO - ================================================================
2023-09-06 04:15:18,176 - INFO - [-] Targeting TrendMicro
2023-09-06 04:15:18,181 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:15:32,888 - INFO - ----------------------------------------------------------------
2023-09-06 04:15:32,934 - INFO - [+] Safety Rating is rated as Safe
2023-09-06 04:15:32,941 - INFO - ----------------------------------------------------------------
2023-09-06 04:15:32,982 - INFO - [+] Category for URL [redacted] is Health

DevTools listening on ws://127.0.0.1:8298/devtools/browser/0df104b3-e0d2-4c0e-81ef-6f97e1150c08
2023-09-06 04:15:45,447 - INFO - ================================================================
2023-09-06 04:15:45,454 - INFO - [-] Targeting McAfee
2023-09-06 04:15:45,458 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:16:10,369 - INFO - ----------------------------------------------------------------
2023-09-06 04:16:10,418 - INFO - [+] Category for URL [redacted] is
2023-09-06 04:16:10,426 - INFO - ----------------------------------------------------------------
2023-09-06 04:16:10,464 - INFO - [+] Reputation for URL [redacted] is Unverified

DevTools listening on ws://127.0.0.1:8483/devtools/browser/71201edf-02fb-4691-8843-1c422b7f4159
2023-09-06 04:16:22,948 - INFO - ================================================================
2023-09-06 04:16:22,954 - INFO - [-] Targeting Lightspeed Systems
2023-09-06 04:16:22,958 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:16:37,781 - INFO - ----------------------------------------------------------------
2023-09-06 04:16:37,787 - INFO - [+] Category for URL [redacted] is security

DevTools listening on ws://127.0.0.1:8702/devtools/browser/344ad282-7e42-4aaa-8f3a-e568fa6b6ddc
2023-09-06 04:16:50,274 - INFO - ================================================================
2023-09-06 04:16:50,280 - INFO - [-] Targeting Brightcloud
2023-09-06 04:16:50,284 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:17:10,166 - INFO - ----------------------------------------------------------------
2023-09-06 04:17:12,269 - INFO - [+] Category for URL [redacted] is   - Health and Medicine
2023-09-06 04:17:12,274 - INFO - ----------------------------------------------------------------
2023-09-06 04:17:12,309 - INFO - [+] Reputation for URL [redacted] is - Moderate Risk (50 of 100)

DevTools listening on ws://127.0.0.1:8895/devtools/browser/25726615-4af0-4fac-887d-729a0bc537b1
2023-09-06 04:17:24,826 - INFO - ================================================================
2023-09-06 04:17:24,832 - INFO - [-] Targeting PaloAlto
2023-09-06 04:17:24,837 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:17:46,845 - INFO - ----------------------------------------------------------------
2023-09-06 04:17:46,850 - INFO - ----------------------------------------------------------------
2023-09-06 04:17:46,886 - INFO - [+] Category for URL [redacted] is Categories: Health-and-Medicine

DevTools listening on ws://127.0.0.1:9196/devtools/browser/4a4b7314-e8f6-4894-a2a2-60c4483594e8
2023-09-06 04:17:59,392 - INFO - ================================================================
2023-09-06 04:17:59,399 - INFO - [-] Targeting BlueCoat
2023-09-06 04:17:59,404 - INFO - [-] Checking category for URL [redacted]
2023-09-06 04:18:11,213 - INFO - ----------------------------------------------------------------
2023-09-06 04:18:11,218 - INFO - [+] Category for URL [redacted] is Not yet rated
```

Domain Reputation ?





Domain Reputation ?

This is an automated response to your review request submitted on 8/2/2023 4:03 AM (CST) for [redacted]

Review time: 8/2/2023 7:29 AM
Original category: parked
Updated category: family.health
Review reason: Hello,

I am writing to introduce my website, [redacted] and express my belief that it should be categorized as Health.

I kindly request your expert assessment of my website and its placement in the appropriate category on your platform. By doing so, you would contribute to its visibility and enable users to find it more easily.

Thank you for your attention. I eagerly await your response.

Categorization reason: Manually moved to family.health by LSSDB\cmasiel at 8/2/2023 7:29 AM CST

If the category has not been changed the content categorization team has determined that the site is categorized correctly in accordance with our published category descriptions. If you still feel you need to access this site please contact your local system administrator.

* Depending on the configuration of your local system, this update may take anywhere from 1 to 24 hours to reach you.

Lightspeed Systems
Content Categorization Team

Disclaimer: This message, including any attachments, is confidential, may be legally privileged, and is intended for the use of the intended recipient. It is the property of Lightspeed Solutions, LLC (dba Lightspeed Systems). If you have received this message in error, please notify us immediately by reply email, or by email to mail.admin@lightspeedsystems.com, and delete this message, along

18:31

URL Classification Change Request -- Support Ticket Number: 841383 Inbox

B **BrightCloud Data Update** 18:20
to me, wr-dbchange

Hello again -

We have reviewed [redacted], and have updated the site to the Health and Medicine categories per your suggestion. This change is now published in the BrightCloud Service and is available in Database version 8.678.

Thanks again for your suggestion!

- Webroot BrightCloud Threat Intelligence Support
Questions? Suggestions? Need help? Contact us at: wr-dbchange@opentext.com

Domain Reputation ?

Lookup data results for Domain

[redacted]

Search by IP, domain, or network owner for real-time threat data.

IP & Domain Reputation Overview | File Reputation Lookup | Email & Spam Data

OWNER DETAILS

DOMAIN [redacted]

CONTENT DETAILS

CONTENT CATEGORY Health and Medicine

Think these category details are incorrect?

REPUTATION DETAILS

WEB REPUTATION — Neutral

Submit Web Reputation Ticket

BLOCK LISTS

TALOS SECURITY INTELLIGENCE BLOCK LIST

ADDED TO BLOCK LIST No

Log in

Test A Site

Enter a domain or URL into the search engine to view details about its current URL categories. To request recategorization of this website, click Request Change below the search results.

URL Enter a URL SEARCH

URL [redacted]

Categories: Health-and-Medicine

Risk Level: Low-Risk

Category: Health-and-Medicine

Description: Sites containing information regarding general health information, issues, and traditional and non-traditional tips, remedies, and treatments. Also includes sites for various medical specialties, practices and facilities (such as gyms and fitness clubs) as well as professionals. Sites relating to medical insurance and cosmetic surgery are also included

Extra ?

Gophish + Evilginx setup:

- Migrate Gophish from SQLite to MySQL
- Configure Evilginx to be able to access the database (deny access for anybody else)
- Make sure Gophish sends the RId to Evilginx
  - ✓ Parse RId and store it in the **Session** struct
  - ✓ Use RId to find the corresponding campaign ID and user email address
- Implement code to connect to MySQL, fetch and update entries in the tables
- Update **results** table as if credentials were submitted
- Update events table to provide details regarding the data captured
  - ✓ Username
  - ✓ Password
  - ✓ User-Agent
  - ✓ Capture Time
  - ✓ IP address
  - ✓ Session Cookie (not yet)

Next iterations:

- Store session cookie in Gophish database + modify UI to include it
- Add CAPTCHA/Cloudflare Check on landing page to avoid it being discovered
- Ignore Gophish metrics if email/link was opened by an automated mechanism

Demo ?

- 1. How to properly create a campaign in Gophish
- 2. How effective is “Blocking the Internet” ?
- 3. What is the Evilginx workflow and how does it integrates into Gophish ?

References ?

- https://twelvesec.com/2023/11/22/the-current-state-of-phishing-attacks/
- https://rastamouse.me/infrastructure-as-code-terraform-ansible/
- https://www.ired.team/offensive-security/red-team-infrastructure/automating-red-team-infrastructure-with-terraform
- https://github.com/fin3ss3g0d/evilgophish

Thank you !

www.twelvesec.com
hello@twelvesec.com