Red Team Tutorial: Design and setup of C2 traffic redirectors

Dmitrijs Trizna · Follow

8 min read · Jan 2, 2021

Abstract

Control of compromised machines within the target network happens through specifically designed *Command & Control (C&C, C2)* infrastructure. This article describes the rationale behind C2 design decisions and provides a step-by-step setup of the C2 redirector.

The report may be valuable for defensive analysts as insight in operations of adversary groups, as well as provide the necessary information for those willing to simulate adversary C2 channels. We will cover how one can build a HTTPS redirector using basic tools — *nginx, minimalistic VPS, free DNS, and PKI certificate services*.

Introduction

C2 infrastructure is built with the intent to pursue several goals:

1. hide the true location of the C2 server;
2. mimic legitimate communication;
3. allow only malware control traffic to reach the real C2 server;
4. be reliable — given detection the part of C2 infrastructure, still, maintain C2 channel to the target.

Simple port forwarding by tools like *socat* or SSH can solve bullet #1 and partly #4. However, to address bullets #2 and #3 we need to introduce more sophisticated redirectors — hosts, which act as reverse proxies to forward only specific traffic to the real C2 server, whilst serving counterfeit content for the arbitrary visitor. In this article, we will focus on HTTPS as a protocol for external C2 communication. A high-level overview of such design is visualized in Figure 1 below:

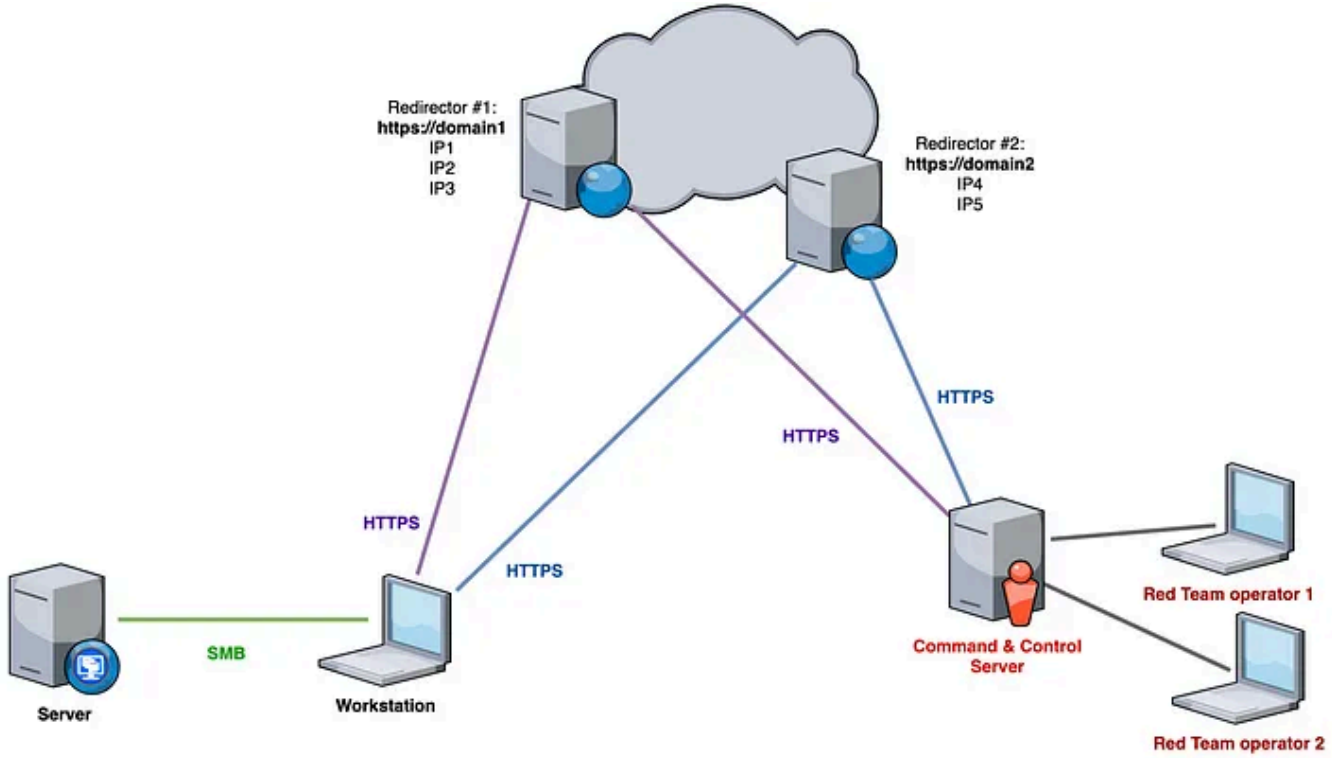

Figure 1. Functional real-world Command & Control infrastructure.

Software setup

The most valuable resource for any redirector is a public IP address, whilst it does not require any somehow powerful CPU or memory resources to achieve its goals. Therefore, minimalistic images from common cloud providers like AWS or Azure are totally fine (e.g. **t3a.nano** from AWS with 2 vCPUs, 500 MB RAM, and 8 GB disk).

After access to such host, we're ready to build our *nginx* setup:

```
sudo yum -y install git nginx vim certbot tmux wget

# SELinux configuration for Reverse Proxy functionality
setsebool -P httpd_can_network_connect true

systemctl restart nginx
```

A few settings might be adjusted before nginx restart in *nginx.conf*, under *http* clause, to support specifics of HTTP traffic, like potential long C2 domain names or file transfers:

```
types_hash_max_size 2048;
server_names_hash_bucket_size 128;
client_max_body_size 128M;
```

Domain and certificate

For Proof of Concept needs, we will use one of the services that allow you to define your DNS subdomain configuration without the need to register your domain, freeDNS [1]. Here, you can register your own A, AAAA, TXT records

under some appealing domains, like *allowed.org, ignorelist.com, awiki.org*, etc.

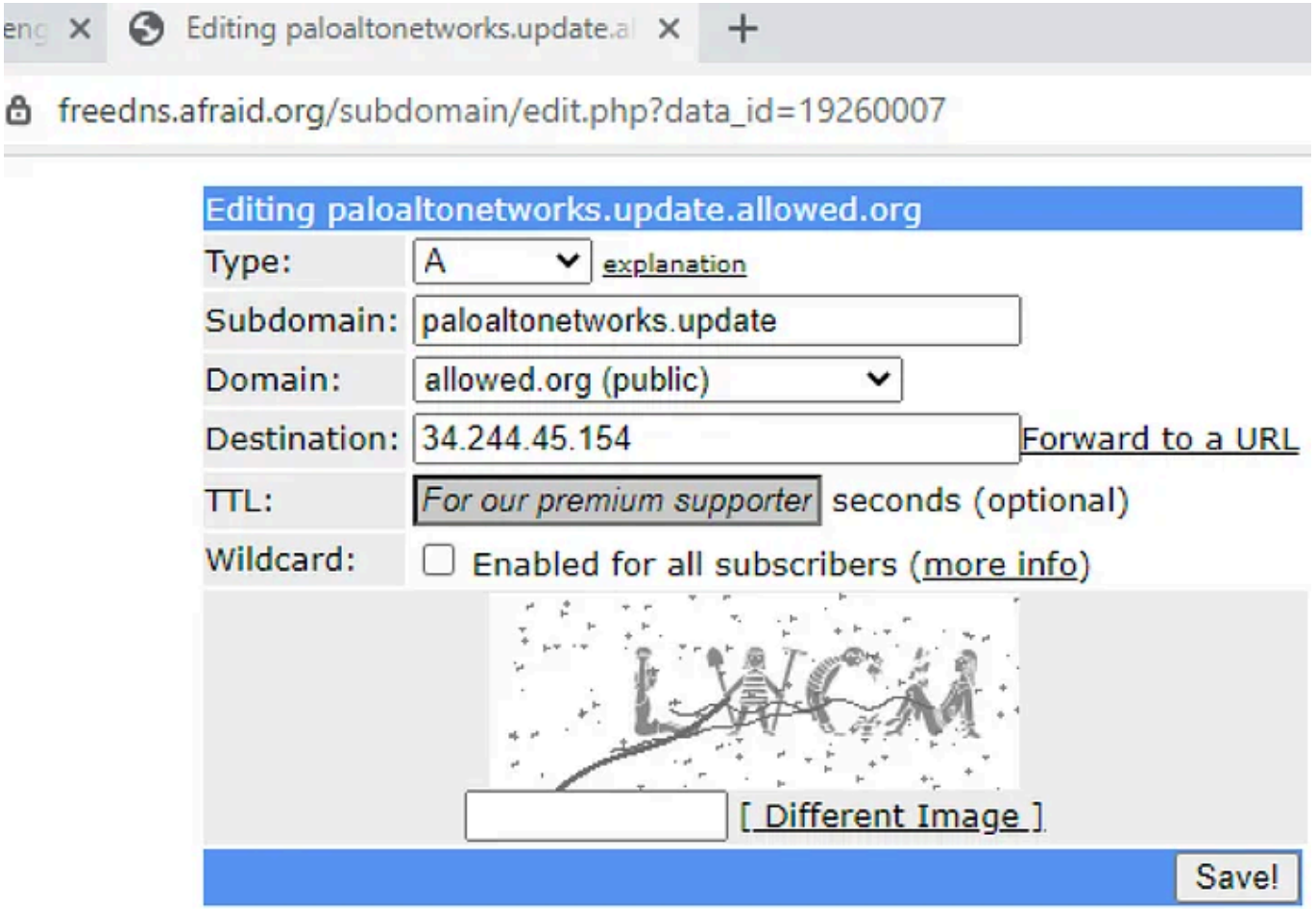

Figure 2. Creating own subdomain A record and pointing it to AWS.

> *Be aware that some mature environments are strict on behalf of dynamic DNS services like these! For such targets, it is always beneficial to have a pre-purchased*

> *domain with a long and clean history. Professional red teams hold multiple domains for redirecting needs, pointing to legitimate services until the actual engagement, whereas real threat actors may use compromised web servers.*

After the newly registered subdomain is distributed, and we see the content of your */usr/share/nginx/html/* behind it, time to request a HTTPS certificate for our domain using *certbot* as follows:

```
certbot certonly --webroot -w /usr/share/nginx/html/ -d
paloaltonetworks.update.allowed.org -m myemail@yahoo.com --agree-tos
-n
```

This will create HTTPS certificates under */etc/letsencrypt/live/<domain>/.*

Proxy Conditions

Now we are ready to setup site configuration in nginx. Before that, it might be beneficial to discuss the design decision for bullet Nr. 3. from the introduction's list: "*allow only agent control traffic to reach real C2 server*".

During an ethical adversary simulation, your main objective is not to harm any involved third party and stay within the bounds of legal agreement. As a result, it is your responsibility to mitigate the compromisation risk of unauthorized machines, for example, if an employee opens an infected macro at his home PC. Moreover, besides actual C2 functionality, the C2 server may host post-exploitation tools, therefore, reveal additional information about the attacker.

Consequently, this is a crucial opsec consideration to control what we let in. Few techniques might be useful here:

- Request or identify (e.g. public ASN records) IP address scope of target services; establish C2 channel only to requests coming from this scope.
- Only requests with specific User-Agent substring are allowed to pass.

In addition to the mentioned benefits, the former helps to prevent the full detonation of agents in vendor sandboxes, latter — protect against defensive analyst actions. To be honest, there's no way to surely hide a real C2 server from clever threat hunters, but these two techniques severely reduce or at least delay such scenario. However, mature opsec should be complemented

by the cleanup of C2 exposure and clever orchestration of redirectors (shutdown of identified proxies).

Nginx configuration below:

- creates HTTPS page with registered subdomains and certificates, serving the content of */opt/paloaltonetworks.update.allowed.org/*;
- proxies traffic to *localhost:8080* only if a request contains substring “41.0.2228.0” and comes either from 123.123.123.0/24 or 213.213.213.0/24;

```
server {
  listen 443 ssl http2 default_server;
  listen [::]:443 ssl http2 default_server;

  server_name paloaltonetworks.update.allowed.org ;
  root /opt/paloaltonetworks.update.allowed.org/;

  ssl_certificate "/etc/letsencrypt/live/paloaltonetworks.update.allowed.org/cert.pem";
  ssl_certificate_key "/etc/letsencrypt/live/paloaltonetworks.update.allowed.org/privkey.pem";
  ssl_session_cache shared:SSL:1m;
  ssl_session_timeout 10m;
  ssl_ciphers HIGH:!aNULL:!MD5;
  ssl_prefer_server_ciphers on;

  location / {
    set $C2 "";
    if ($http_user_agent ~ "41.0.2228.0") {
      set $C2 A;
    }
    if ($remote_addr ~ "123.123.123") {
      set $C2 "${C2}B";
    }
    if ($remote_addr ~ "213.213.213") {
      set $C2 "${C2}B";
    }
    if ($C2 = "AB") {
      proxy_pass https://localhost:8080;
    }
    try_files $uri $uri/ =404;
  }

  error_page 404 /404.html;
  location = /opt/html/40x.html {
```

```
    }
    error_page 500 502 503 504 /50x.html;
    location = /opt/html/50x.html {
    }
}
```

c2-nginx-site.conf hosted with ❤ by GitHub view raw

On C2 server, start HTTPS listener on any arbitrary port, e.g. 443, and forward this port from redirector locally:

```
ssh user@redirector -R 127.0.0.1:8080:127.0.0.1:443 -f -N
```

We deliberately skip the C2 server setup part in this article. This is a separate topic, there are many C2 servers, and, therefore, variability available [7]. Few notes here:

- Be sure to point agent callback to your subdomain, not C2 listener bind-address;
- If the C2 server requires PKCS12 HTTPS certificate format, bundle it from letsencrypt files using *openssl*. Example for Covenant C2 framework:

```
openssl pkcs12 -export -in fullchain.pem -inkey privkey.pem -out
certificate.pfx -name paloaltonetworks.update.allowed.org -passout
pass:CovenantDev
```

Webpage imitation

Contemporary networks rely on domain and website reputation analysis. For example, if a domain is known to distribute malware, all responsible proxy and firewall vendors will apply this knowledge to prevent any of their customer users to visit this webpage.

The same categorization logic helps to control user traffic to a social network, gambling, or pornography pages. It is important for your redirector to fall in some legitimate category during an exploitation phase, like business or education. Moreover, some restrictive environments (like fintech) may introduce whitelisting by these categories.

Initial categorization of new pages mostly happens using machine learning models with multi-label classification and is a well-published topic [2][3]. Input features are formed using page meta-information — title, specific words in a text body, javascript behavior, etc.

The best way to mimic a legitimate business page and fall into the right category is to clone a business page. Clever imitation may work well even if a security analyst manually visits your page during incident response — if page contents and domain represent their infrastructure functionality well, the C2 channel may stay undetected, and therefore active, for a prolonged period of time.

Subdomain we registered presumes update information for any Palo Alto Networks (PAN) product, so we chose to clone the appropriate page under *docs.paloaltonetworks.com*. *wget* serves these purposes well, but be sure to use *tmux* session or *nohup*, to detach the download process from an SSH session, as it may take a while:

```
mkdir /opt/paloaltonetworks.update.allowed.org && cd !$

nohup wget --limit-rate=200k --reject pdf --no-clobber --convert-
links --random-wait -r -p -E -e robots=off -U mozilla
https://docs.paloaltonetworks.com/resources/recent-release-note-
updates.html &
```

Figure 3. Imitation of PAN documentation on rogue subdomain.

If your target page does not display properly, you may consider adding the following *wget* options during the clone:

```
--recursive
--no-parent
--page-requisites
--adjust-extension
```

After a while, we can verify the categorization of our page across major vendors: PAN [4], Symantec (i.e. BlueCoat proxy)[5], CheckPoint [6], etc.

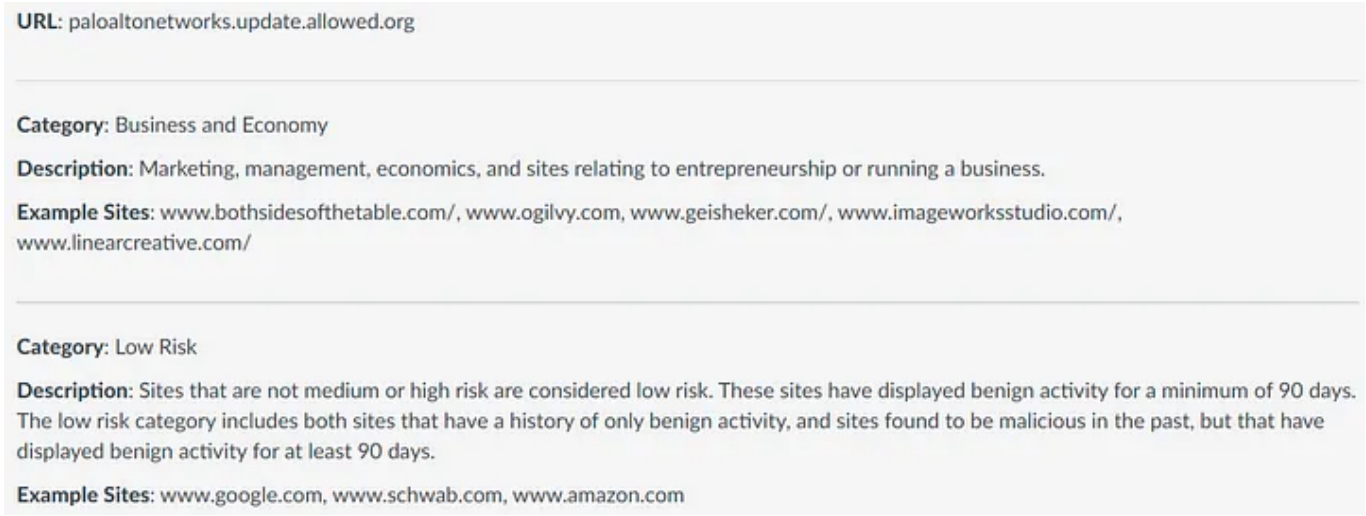

Figure 4. C2 redirector categorization by PAN.

If the target's proxy is known or can be identified from past vacancy descriptions, employee LinkedIn profiles, or public purchase records, it is crucial to target specific vendor opinion.

Hauls

At this point, our C2 infrastructure is fully functional and can be used for real-world adversary simulation. Although, all the previous work does not address bullet Nr. 4. from the initial list: "*be reliable — given detection the part of C2 infrastructure, still, maintain C2 channel to the target*". This subsection does not cover any additional settings, but raises a discussion on how real-world adversaries achieve this requirement.

As you may recall, Figure 1 represents two redirector groups, hidden behind different domains and IP addresses. One of these is used for interactive manipulation of compromised resources for lateral movement and data exfiltration (in professional literature is called Short Haul [8]), whereas the second acts as a passive callback channel and Long Haul.

Long Haul relies on a separate network channel (e.g. may use DNS tunneling instead of HTTPS transport) and may utilize different malware family or persistence technique. Agents behind this channel may have rare callbacks (could be once in a month) and are used only to restore Short Haul.

Therefore, security analysts should consider the detection of the backup channel, with a completely different network fingerprint. Any connection from a compromised host to an unauthorized host should be under suspicion.

For adversary simulation operators, all work described above should be done at least twice. It is possible to use the same C2 server for the Long Haul channel, but mature adversaries use separate infrastructures for both hauls.

Conclusions

In this article, we discuss key concepts of real-world adversary Command & Control infrastructure. With this knowledge, security engineers have insight into the threat actors' approaches and methods, consequently possess better incident response capabilities. We propose a simple C2 redirector design, for other ethical red team groups to borrow this approach. That should result in a better simulation of adversary groups, and provide more valuable training for defensive infrastructures.

References

[1] (Jan. 2021), "Free DNS — Dynamic DNS — Static DNS subdomain and domain hosting", [Online], Available: https://freedns.afraid.org

[2] P.D. Woogue, G.A.A. Pineda, C.V. Maderazo, "Automatic Web Page Categorization Using Machine Learning and Educational-Based Corpus", International Journal of Computer Theory and Engineering, 2017.

[3] F.D. Fausti, F. Pugliese, D. Zardetto, "Toward Automated Website Classification by Deep Learning", 2019.

[4] (Jan. 2021), "Palo Alto Networks URL filtering — Test A Site", [Online], Available: https://urlfiltering.paloaltonetworks.com

[5] (Jan. 2021), "Symantec Sitereview", [Online], Available: https://sitereview.bluecoat.com

[6] (Jan. 2021), "URL Categorization | Check Point Software Technologies",

[Online], Available: https://urlcat.checkpoint.com/urlcat/
[7] (Jan. 2021), “The C2 Matrix”, [Online], Available: https://www.thec2matrix.com/matrix
[8] (Sep. 2014), “Infrastructure for Ongoing Red Team Operations”, [Online], Available: https://blog.cobaltstrike.com/2014/09/09/infrastructure-for-ongoing-red-team-operations/

Cybersecurity Networking Nginx DNS

Written by Dmitrijs Trizna

396 Followers · 28 Following

Security Researcher @ Microsoft. This blog is an independent R&D at the intersection of Machine Learning and Cyber-Security.

Follow