Detection Systems for Changes in an Organizations Supply Chain

A lot of people don't know what a supply chain is, I know I didn't when I started this blog, so I decided to create this blog to help others get familiar with what they are and how to secure them and manage them.

Throughout this blog I hope to give like-minded security researchers an insight into what supply chains are, how they work in an organization, how to use OSINT to map out external perimeters and how to detect system changes that could be malicious.

What is a supply chain?

"A supply chain is a system of organizations, people, activities, information, and resources involved in supplying a product or service" - Investopedia

In better terms: A supply chain is the entire organization as a whole as well as anything the organization links to (subdomains, third parties etc)

Attackers hunt for unsecure network protocols, unprotected server infrastructure, or bad coding practices. They could use compromised developer accounts, stolen certificates, or unsafe hardware devices to exploit other components of the supply chain of an organization.

An example of a supply chain for Netflix would be: Hosting on servers, services, applications, APIs, resources (the videos and other content), encoding the content, customers, customer support, employees and more.

Cameron Noakes

Supply Chains allow organizations to understand their business layout and what they rely on as an organization to provide a service or product to their customers, it is a general view of how everything links together inside a company.

Depending on the component in the supply chain there can be various attack surfaces, one targeting Application programming interfaces (APIs) could potentially exploit APIs or subdomain logins or any forms of web app exploitation, whereas targeting the people of the organization through phishing would cause different outcomes due to the different attack vectors.

If an attacker cannot exploit an API, they could try phishing on a customer or employee of the company, if failure is persistent then they could move onto injecting malware onto one of the organizations servers or even trick their processes to respond to an attackers server hosting malware to distribute it further down the line to customers of the organization. This is what happened to Shylock:

"The Shylock attackers compromised legitimate websites through website builders used by creative and digital agencies. They employed a redirect script, which sent victims to a malicious domain owned by the Shylock authors" - https://www.ncsc.gov.uk/collection/supply-chain-security/website-builders

The possibilities of potential attacks can be from anywhere in an organizations supply chain, which is why adequate security needs to be put in place in all sections of a company's supply chain.
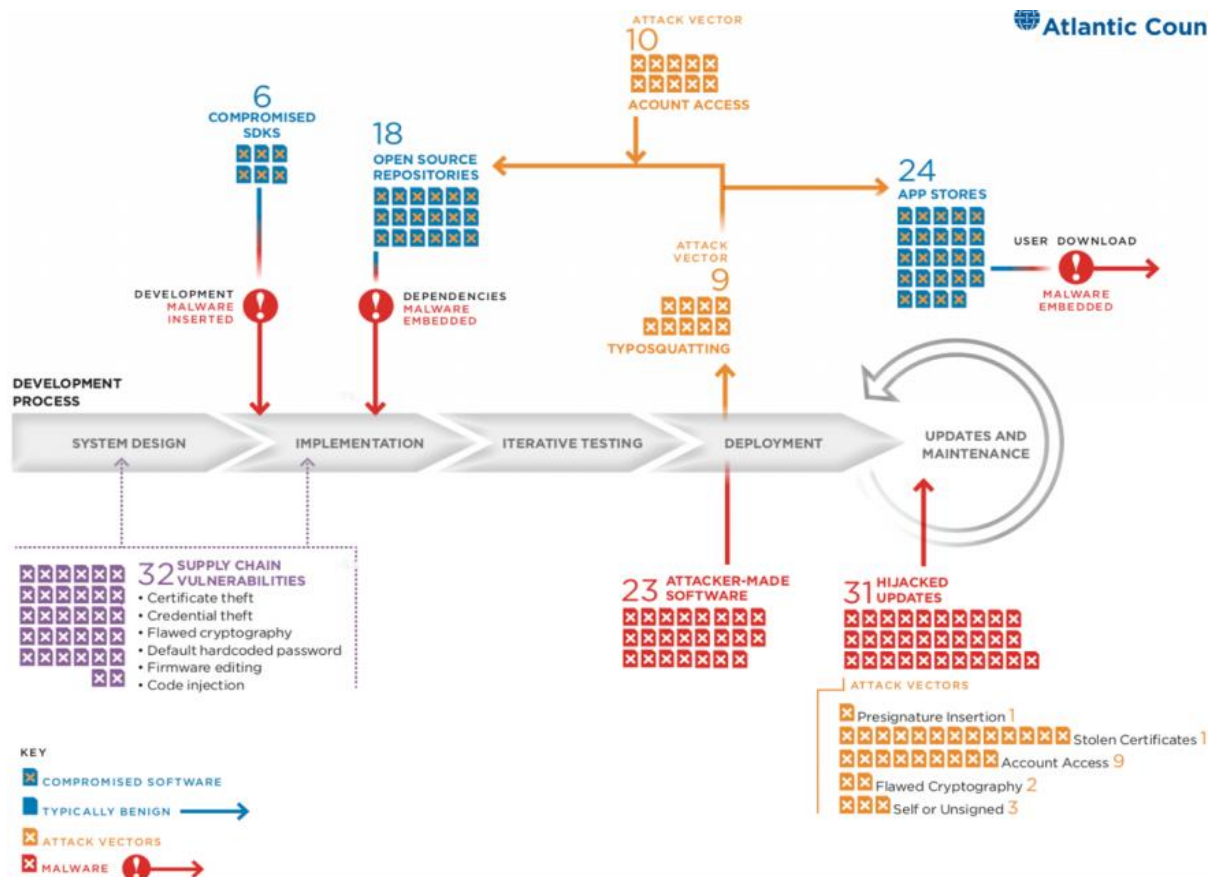
First of all, we need to obtain some 'seed data', information about a target organization to kick off the Open Source Intelligence (OSINT) gathering. Seed data is needed first in order to start reconnaissance of an organization.

Overview

As this blog post is more focused on changes that can be made from an attackers perspective for a supply chain we will be more geared towards changing source code and adapting it and manipulating it to suit the attackers needs, examples could be to

Cameron Noakes

execute malicious applications, respond to an attackers server as well as bypass validation and verification. These types of attacks are called 'Software Supply Chain Attacks' as they specifically target software made by or used for the organization.

A clear view of a general software organizations supply chain can help to get an overview of how an attacker would go about exploiting parts or components of a supply chain.

https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/#introduction

From the above diagram you can individually see each section of a basic supply chain up to the customer with various attack vectors, the job title 'Supply chain manager' helps to organise the consistent flow of products or services to the customer, they also can be responsible for security the supply chain by potentially outsourcing to various security companies to limit the attack surface and minimise the risk of impact.

The types of organizational impact of attackers

There are many ways an organization could be impacted by attackers targeting their supply chain, a few examples are listed below but tend to link with one another.

- Reputational damage – customers do not trust the company

Cameron Noakes

- Financial impact – loss of finance
- Data disclosure – attackers leak files and data
- Customer impact – customers are attacked, and data is disclosed/ compromised
- Clear up – the organization may have to deal with lawsuits and further attacks

Sections like customer impact tie in with reputational damage and clear up due to the customer losing faith in the company and in return could result in potential lawsuits from customers due to them being targeted and not adequate security measures in place.

A supply chain is vast and extensive research needs to be done in order to map out organizations and domains, one organization may outsource to a third party for security software, and the security software company may outsource to another company, following down the supply chain you can map out an external perimeter of the original organization attackers wanted to target.

Any vulnerabilities found on third party domains and companies could directly impact the target company due to them being a third party.

MAIN CONTENT

For the main aspect of this blog we will use Barclays Bank to run passive and active reconnaissance on to show more examples of how supply chains could be exploited. Keep in mind that only basic active reconnaissance will be taking place such as DNS lookup and subdomain enumeration as we do not want to actually target Barclays Bank.

"With nearly 16,000 companies from more than 41 countries supplying us across a broad range of products and services, our supply chain helps us deliver for all our customers, clients and colleagues." – Barclays

The bigger an organization gets, the bigger attack surface over their supply chain increases. As Barclays is a British multinational investment bank and financial services company, it has a very large-scale supply chain, making it easier for attackers to compromise components of the supply chain but in Barclays defence they have more funding for cyber security.

The current seed data for Barclays would be their domain:

The website for Barclays is www.barclays.co.uk

Cameron Noakes

This website will be our starting point for OSINT, it will give us a target to work with, a domain that could contain third parties.
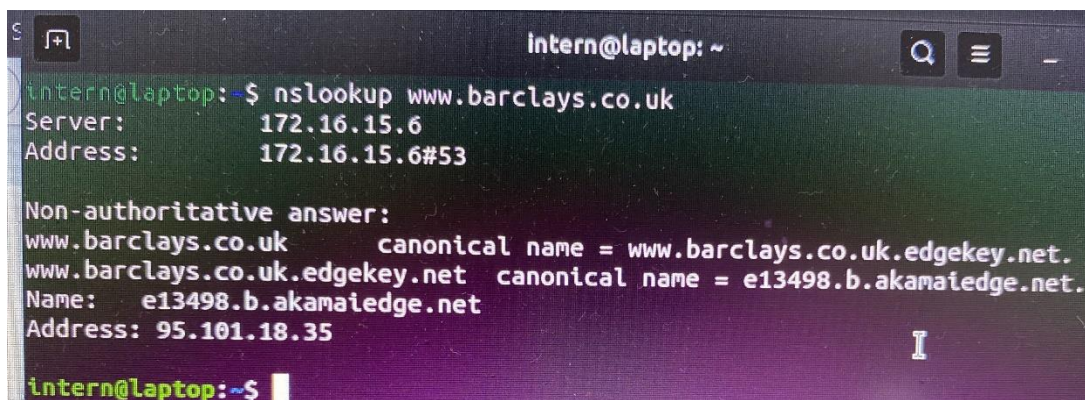
Website Tooling for Mapping (APIs, frameworks, subdomains):

We can try to find out how everything interacts with their main website by looking at source code, running tools against it and just having a general look around.

Nslookup:

Nslookup is a command-line Linux tool that allows querying of data regarding a website. It can return information such as frameworks used and IP addresses of points of interest.

When you run this through nslookup you get the .net frameworks which can help you map out the start of the supply chain of Barclays by finding what it relies on.

Dig:

Dig is another command-line tool much similar to nslookup it is used for querying website information.

I complemented this nslookup with dig to make sure the information was accurate. The dig query returned the same two .net frameworks proving they are used but gave a different server IP, I'd imagine this is due to Barclays being a big organization thus having many servers running processing different requests or is a use of 'false positives'.

This type of output would be useful to map out frameworks and potential subdomains Barclays uses as well as reference points to how everything interacts on their website.

Subbrute.py

Subbrute.py is a python script that runs through a series of lists containing commonly used subdomain path names to identify links between that of the main website.

No new information was discovered using this tool, returns the two main frameworks used and that was it.

Whois

A 'whois' command returns valuable information about the host, in this case the host is the main domain of Barclays, just like the other lookup queries. The whois query of Barclays can be seen below:

Cameron Noakes

The Barclays Domain was registered by a company called 'netnames.co.uk', we can add this as a third party of Barclays, we will continue this process with each successive third party until a mapped perimeter is near complete.

NetNames

NetNames is a British organization that provides online brand protection, as well as domain name management and acquisition services.

NetNames is a third party of Barclays and potentially handles the domain and brand security as well as the domain registration, this is because Barclays is a financial firm and not a cyber security one thus outsourcing to a cyber security company for security is advantageous as the security will be of a higher standard and takes responsibility off of Barclays.

Whois on NetNames

Running a whois on netnames.com returns lots of information on location as well as emails and admin names but also expiry and creation dates of the domain, which makes the OSINT gathering more helpful for mapping out Barclays perimeter through NetNames.

Cameron Noakes

```
          : $ whois netnames.com
Domain Name: NETNAMES.COM
Registry Domain ID: 4106367_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.corporatedomains
Registrar URL: http://www.cscglobal.com/global
Updated Date: 2020-02-18T13:32:55Z
Creation Date: 1995-09-04T04:00:00Z
Registry Expiry Date: 2020-09-03T04:00:00Z
Registrar: CSC Corporate Domains, Inc.
Registrar IANA ID: 299
Registrar Abuse Contact Email: domainabuse@cscg
Registrar Abuse Contact Phone: 8887802723
Domain Status: clientTransferProhibited https:/
Domain Status: serverDeleteProhibited https://
Domain Status: serverTransferProhibited https:/
Domain Status: serverUpdateProhibited https://
Name Server: UDNS1.CSCDNS.NET
Name Server: UDNS2.CSCDNS.UK
```



```
Registry Registrant ID:
Registrant Name: Domain Administrator
Registrant Organization: CSC Corporate Domains, Inc.
Registrant Street: 251 Little Falls Drive
Registrant City: Wilmington
Registrant State/Province: DE
Registrant Postal Code: 19808
Registrant Country: US
Registrant Phone: +1.3026365400
Registrant Phone Ext:
Registrant Fax: +1.3026365454
Registrant Fax Ext:
Registrant Email: admin@internationaladmin.com
Registry Admin ID:
Admin Name: Domain Administrator
Admin Organization: CSC Corporate Domains, Inc.
Admin Street: 251 Little Falls Drive
Admin City: Wilmington
Admin State/Province: DE
Admin Postal Code: 19808
Admin Country: US
Admin Phone: +1.3026365400
Admin Phone Ext:
Admin Fax: +1.3026365454
Admin Fax Ext:
Admin Email: admin@internationaladmin.com
Registry Tech ID:
Tech Name: DNS Administrator
Tech Organization: CSC Corporate Domains, Inc.
Tech Street: 251 Little Falls Drive
Tech City: Wilmington
Tech State/Province: DE
Tech Postal Code: 19808
Tech Country: US
Tech Phone: +1.3026365400
Tech Phone Ext:
Tech Fax: +1.3026365454
```

Dig on NetNames

Running the dig command on NetNames returns the domain IP address which could potentially be an attack vector for an attacker.

Cameron Noakes

Supply Chain Diagram

From the information we find along the way we can accurately put together a diagram to show how and where everything links to the host of Barclays:

Security on target Organizations are high - Cross-Site Scripting Example

Cross Site scripting (XSS) is when attackers can inject malicious JavaScript code into client browsers. If this was vulnerability, attackers could target the customer side of the supply chain, allowing the attacker to gain access to customer data and banking information.

Cameron Noakes

Barclays makes sure to have a decent level of security for their domain as they are a large company, targeting third parties would be easier to exploit Barclays instead of directly probing it because they would have more vulnerabilities due to them being a smaller target organization.

Do not only target the main organization! – Third Parties

Many big organizations have high security, this tends to happen when a certain organization need to secure data against threats, their security is top of the line because they are a larger scale organization, whereas organizations may not think about third party security as much, as they may think it wasn't much of a big deal and isn't their responsibility. If attackers target a third party of their original target they could more easily exploit the third party which would hold data on the original targeted organization.

For example:

Original target -> company1 (big company, hardened security)

*company1 outsources code to company2* (company2 is smaller and lower security)

By targeting company2 you could get into company1.

Software Supply Chain Attacks:

Attackers can start targeting software to exploit to allow access to data or to modify it in some way to adapt the functionality of the code which could affect third party companies. Software supply chain attacks are becoming more prevalent now as more companies use 'software as a service' (SAAS), making it a larger attack vector.

The graph below shows relevant information on the amount of software supply chain attacks with how many are actually disclosed. These attacks are involving only software supply chain attacks.

Figure 2: Codebase by attack and disclosure

Cameron Noakes
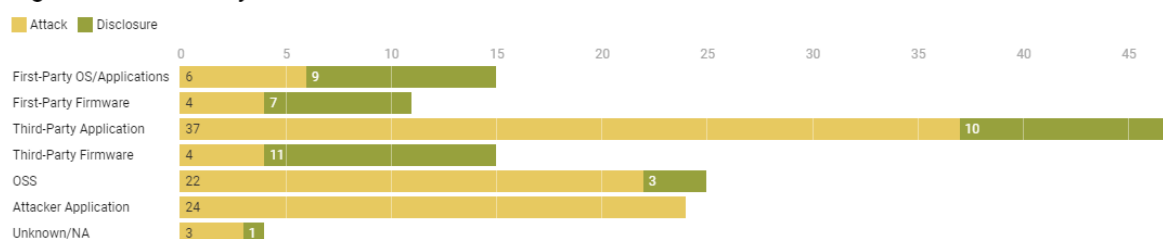
There are far more attacks that haven't been disclosed than ones that have, this identifies that attackers are targeting organizations software to exploit and only a minority of them are being disclosed.

The Barclays Bank website does implement adequate security measures to ensure the company and customer sensitive data is safe, secure and can only be viewed and altered by authorised users.

I performed a reflected Cross-Site Scripting (XSS) attack on the login page and returned explaining how due to my request my session has been terminated. This explains that the Barclays website is not vulnerable to XSS on the login page and developers have been trained correctly to know about these vulnerabilities.

Abusing Code Signing

Code signing allows only verified code to run and more specifically certificates of the code were targeted. Attackers manage to bypass code signing by abusing a variety of factors of certification such as signing their own certificates, the organization using a broken signature system or using a stolen certificate, this allows attackers to run malicious code on victims devices which might look like a trusted source in which could result is reputational damage to the organization.

Developers tend to use a private key taken from the digital Certificate Authority (CA) certificate which as a result can be used to validate and secure code or content. These certificates can be used on proprietary software, open source software (OSS), websites or any content that could cause issue if exploited.

If an organizations architecture for software was a thin-client, attackers could target the software on the server and manipulate and change it.

Automated detection systems would need to be put in place to check the authenticity of the certificate and ensure it is from a trusted source before using it to sign code or verify software otherwise attackers could potentially sign malware with a private key from a certificate they signed themselves and run automated malware on a system.

Cameron Noakes

Detecting Obfuscated Malware

Most malware now is known and can be identified by the file signatures that are scanned my antivirus software when performing security scans, organizations can setup detection systems for servers so that before any software is uploaded to their servers it is scanned before doing so, this ensure that any known malware cannot compromise their servers. However, due to the blue team security researchers, attackers are now having to think of more elaborate ways to get malware onto organizational systems.

A way attackers are doing this is obfuscation. Obfuscated code is code that has been hidden and encrypted in a way to prevent antivirus software knowing the source codes intentions, it is a way to bypass security software as it makes it harder for humans and machines to read the code and interpret it.

There are many ways to make obfuscated code, a simple example is given below:

'javascriptobfuscator.com' is a website that allows users to obfuscate their JavaScript code through URL encoding.

```
message = ("this is a obfuscator test!")

alert(message)
```

The code can now be hidden using this online obfuscator tool, the result is the same code but hidden a bit more so developers and antivirus software might not look into it.

```
var _0x9821=
['\x74\x68\x69\x73\x20\x69\x73\x20\x61\x20\x6F\x62\x66\x75\x73\x63\x61\x74\x6F\
x72\x20\x74\x65\x73\x74\x21'];message=(_0x9821[0]);alert(message)
```

This output could potentially get past checks made before uploading data to a server. This is not malware; this is an example of code obfuscation that can be used when attackers code malware and try to bypass security checks and validation. Organizations could alternatively de-obfuscate any code before it is scanned and uploaded to their servers as a form of detection.

Cameron Noakes

Barclays Third Parties

Barclays uses a company called '**Mobile Iron'** they are a software company which provides endpoint and enterprise mobility management for devices such as multi-factor authentication.

Let us add this to our supply chain diagram to get a better visual representation of this new data:
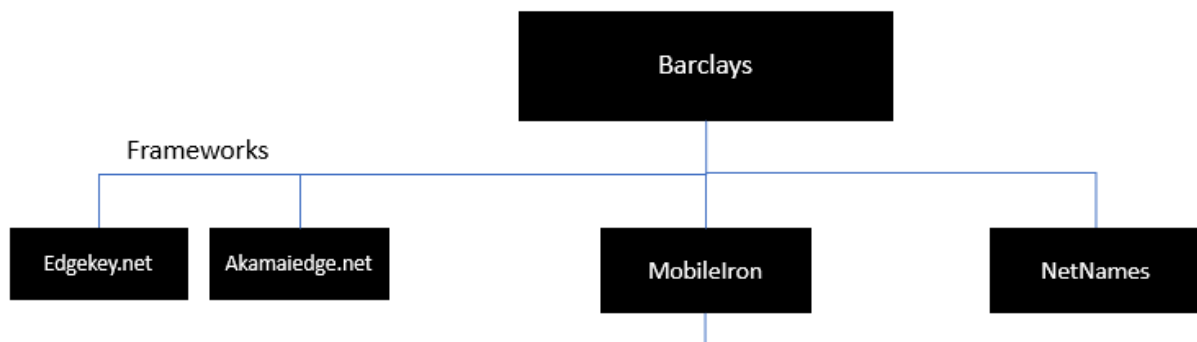
MobileIron

Barclays uses MobileIron as endpoint security as a third party, therefore, it goes without saying that if MobileIron has a vulnerability then so does Barclays, let us get digging.

MobileIron's software allows the security and management of mobile devices such as smartphones and tablet computers within an organizational environment. This allows organizations to secure the digital workspace.

Cameron Noakes
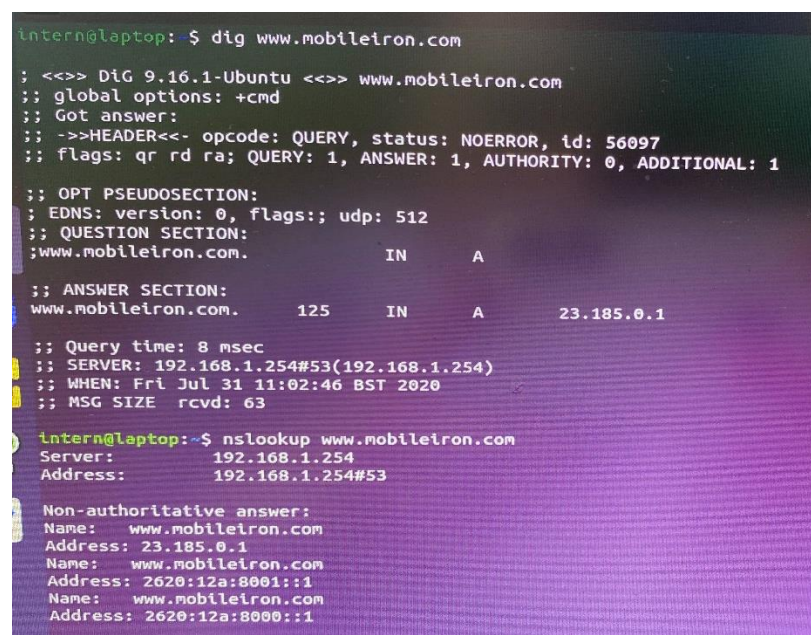
MobileIron reduces the need for passwords by replacing them with biometrics, the hard to manipulate type as you cannot often forge a fingerprint for access.

As I do not have direct access to the MobileIron micromanagement software to try and exploit it, I will have to go about finding vulnerabilities in MobileIron a different way which in turn would still become a vulnerability of Barclays, I am hoping for either information disclosure, subdomains, APIs or frameworks to be found through passive reconnaissance.

Exploring the MobileIron Domain

I performed a dig and nslookup query on MobileIron but found very little (expected as they are a security software company).

Robots.txt

Time to think outside the box. Robots.txt can contain information and directories and files that are meant to be used for Googles crawlers for Search Engine Optimization (SEO) but can be useful for attackers, normally companies secure this page to make sure it cannot be accessed, let's give it a try.

```
#
# robots.txt
#
# This file is to prevent the crawling and indexing of certain parts
# of your site by web crawlers and spiders run by sites like Yahoo!
# and Google. By telling these "robots" where not to go on your site,
# you save bandwidth and server resources.
#
# This file will be ignored unless it is at the root of your host:
# Used:    http://example.com/robots.txt
# Ignored: http://example.com/site/robots.txt
#
# For more information about the robots.txt standard, see:
# http://www.robotstxt.org/robotstxt.html

User-agent: brightedge
Allow: /

User-agent: *
# CSS, JS, Images
Allow: /core/*.css$
Allow: /core/*.css?
Allow: /core/*.js$
Allow: /core/*.js?
Allow: /core/*.gif
```

The 'allows' and 'disallows' are to show the web crawlers where they can index, as we are not web crawlers we can access whatever we want, providing there isn't security blocking us, scrolling down robots.txt there are 'disallows' which tend to be more useful, lets open 'README.txt'.

```
CONTENTS OF THIS FILE
---------------------

 * About Drupal
 * Configuration and features
 * Installation profiles
 * Appearance
 * Developing for Drupal
 * More information


ABOUT DRUPAL
------------

Drupal is an open source content management platform supporting a variety of
websites ranging from personal weblogs to large community-driven websites. For
more information, see the Drupal website at https://www.drupal.org, and join
the Drupal community at https://www.drupal.org/community.

Legal information about Drupal:
 * Know your rights when using Drupal:
   See LICENSE.txt in the "core" directory.
 * Learn about the Drupal trademark and logo policy:
   https://www.drupal.com/trademark


CONFIGURATION AND FEATURES
--------------------------

Drupal core (what you get when you download and extract a drupal-x.y.tar.gz or
drupal-x.y.zip file from https://www.drupal.org/project/drupal) has what you
need to get started with your website. It includes several modules (extensions
that add functionality) for common website features, such as managing content,
user accounts, image uploading, and search. Core comes with many options that
allow site-specific configuration. In addition to the core modules, there are
thousands of contributed modules (for functionality not included with Drupal
core) available for download.
```
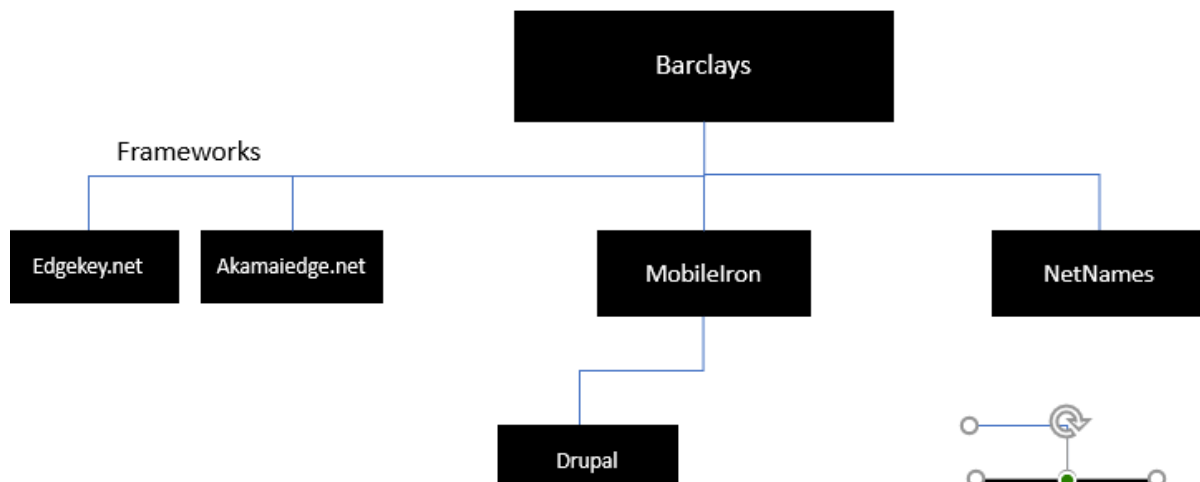
Drupal can be added to the list of third parties for Mobile Iron, Drupal is a software management solutions organization, we will look into them a bit later.

Cameron Noakes

Scrolling through it says to view /core/INSTALL.txt for further information about Drupal websites.
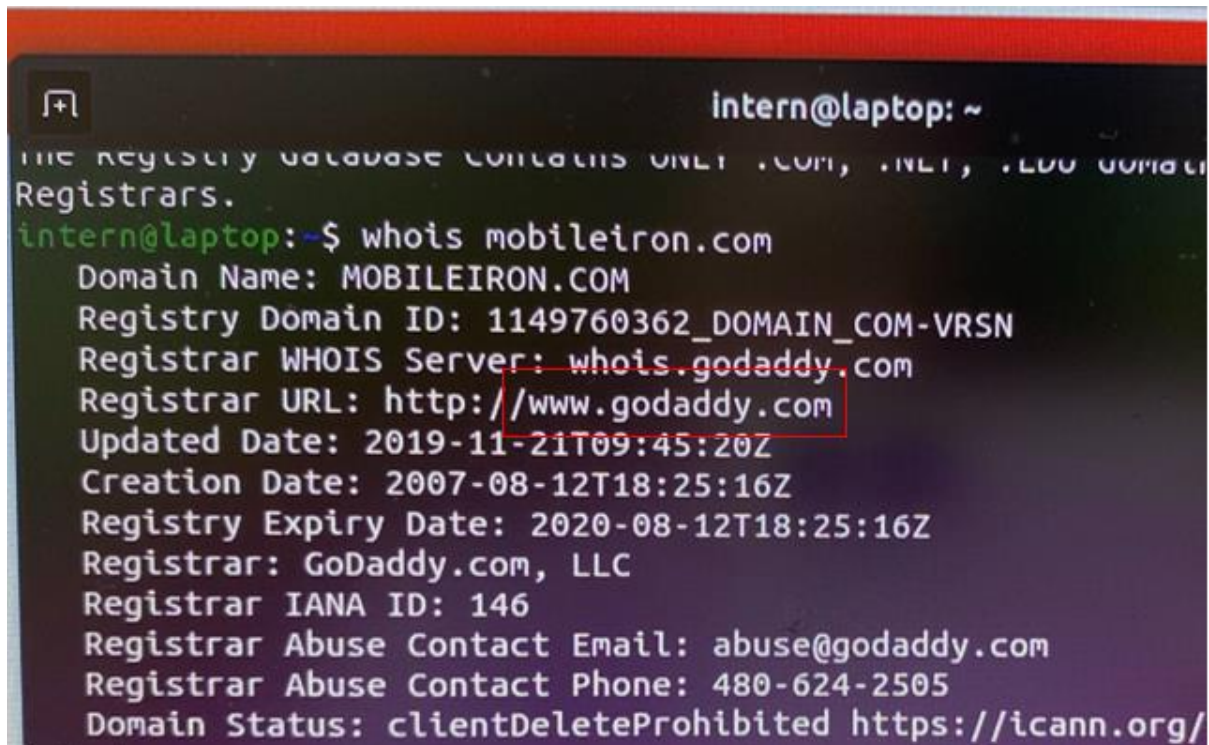

At first, I didn't see any information standing out to me when I saw this, it's just a standard requirements page that comes with customer websites of Drupal, until I realised that in order to have a Drupal website you NEED to have these, therefore, MobileIron must have these too.

From robots.txt you can also access many useful pages on Mobile Irons website such as /admin, /web.config, /README.txt and /INSTALL.txt. These should all be secured as this could account for sensitive information disclosure about Mobile Irons organization and in turn, also could allow information disclosure for their third parties, like Barclays.

The web.config file shows how long cache stays in the browser for Mobile Irons website amongst other semi-relevant information for example securing /.htaccess, unfortunately there was not much in this configuration file.

Running a 'whois'

I remembered that running a 'whois' on the domain can return information about the domain.

Mobile Iron has been registered by GoDaddy.com. We can add this to the third-party list of MobileIron. We can investigate GoDaddy.com for further third-party information, Barclays external perimeter is getting mapped more and more the further I research into it.

C
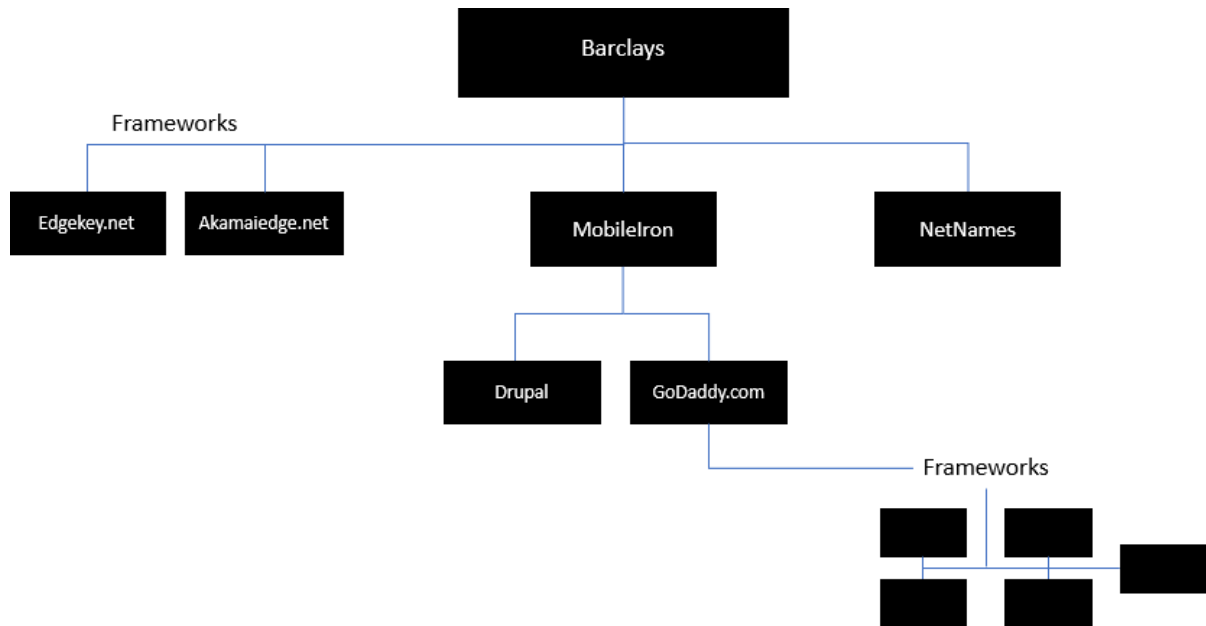
Ripe Text Search

Ripe is a search interface that is an internet regional registrar, it allows querying of information about a specific domain such as the IP address ranges.

Using Ripe I was able to get an IP address range for Mobile Iron, this could be good for mapping out the perimeter of Barclays by seeing if any of these IP addresses have in the past been associated with crypto mining or malware use which could oppose a threat.

Google Dorks Searching for Barclay Subdomains

Sounds cool right? Google Dorks are parameters Google can accept to help find more relevant and related searches, for example if you only wanted to return webpages with

the word 'Netflix' in you could with 'site:Netflix.com'. This technique is very popular with attackers for OSINT and to enumerate specific webpages and subdomains that could be potentially hidden.
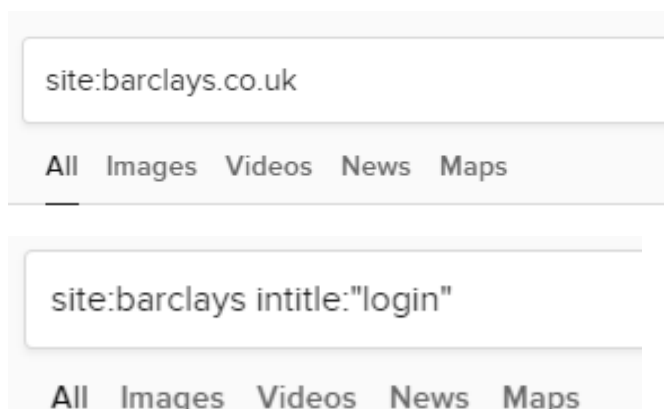
Exploit Database is a publicly available database which tries to store all potential attack components (code, CVEs, exploits, commands, and Dorks).

Using the search bar for the Google dork of 'site:' allows all dorks that specifically search for a domain to be returned. I removed a few search items from the first dork, instead of '**site:admin.*.*/ intext:"login" intitle:"login"'** I used '**site:barclays intitle:"login"'** to make it easier to use, understand and had to replace the placeholders of 'admin' with 'Barclays'. I then plugged it into Google and results were returned.

Running queries with Google dorks for Barclays returns much more than I could have imagined, I was able to enumerate several subdomains tied with Barclays using these two combinations of Google Dorks.

These returned subdomains of the following:

https://gov.identity.barclays

Cameron Noakes

https://labs.uk.barclays/login

https://myapplication.business.uk.barclays/ib/login

https://ledgershare-support.uk.barclays

https://www.apply.barclays.co.uk

https://bank.barclays.co.uk

These results allow us to map out Barclays a lot more as we can see where the Barclays domain branches out to. Let us adapt and continue our supply chain diagram.
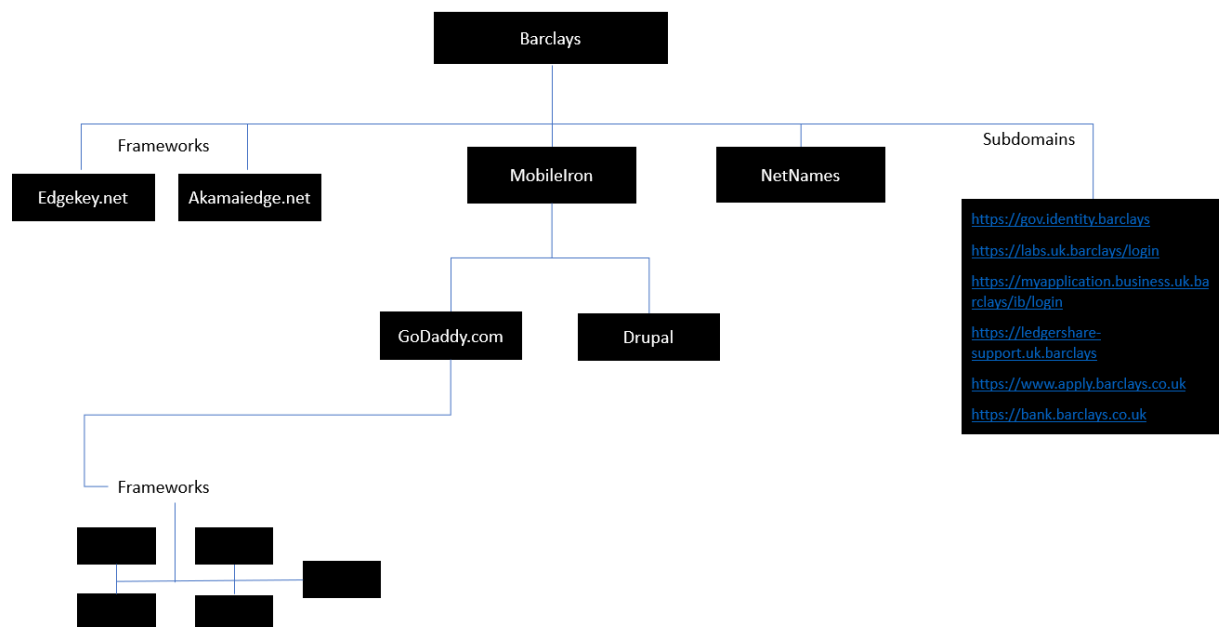
DNSDumpster

Can be used for enumerating subdomains as well as giving the IPs for them, unfortunately using this website tool it did not give back any real domains that Barclays owns, but might work on different target domains.

Crt.sh on Barclays

Crt.sh is a website that allows certificate searches on domains, it can be useful for subdomain enumeration as well as seeing who issued the domain certificates. Each domain will need a new certificate and have to be issued.

Using this web tool, you can deduce who issued the certificates in which can be potential third parties.

Cameron Noakes

The part of the returned list we are interested in is the main domain:

The issuers name can be seen on the far right in which is a direct URL, once visited you find out that a company called 'entrust' issued the certificate as well as the type of encryption used for the certificate and information about entrust, which can be a third party attack vector for targeting Barclays.

```
1671
Subject:
    commonName              = Entrust Certification Authority - L1M
    organizationalUnitName  = (c) 2014 Entrust, Inc. - for authorized use only
    organizationalUnitName  = See www.entrust.net/legal-terms
    organizationName        = Entrust, Inc.
    countryName             = US
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        RSA Public-Key: (2048 bit)
        Modulus:
            00:d0:81:c1:39:23:c2:b1:d1:ec:f7:57:dd:55:24:
            36:91:20:22:48:f7:fc:ca:52:0a:b0:ab:3f:33:b5:
            b0:84:07:f6:df:4e:7a:b0:fb:98:22:3d:01:ac:56:
            fb:71:6d:b2:ee:b9:a0:0f:52:77:ab:98:93:be:33:
            8a:eb:87:5e:c7:aa:b0:ca:69:8f:43:08:6a:3f:22:
            bf:33:39:46:d5:94:f2:e2:4c:05:22:d9:67:80:91:
```
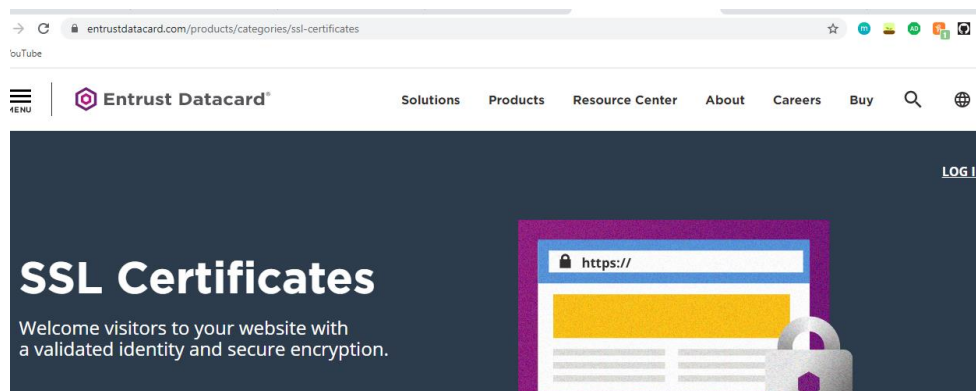

Entrust has a domain for clients to buy the certificates they need. This organization can be added to the supply chain diagram to further map out the Barclays perimeter.

Cameron Noakes
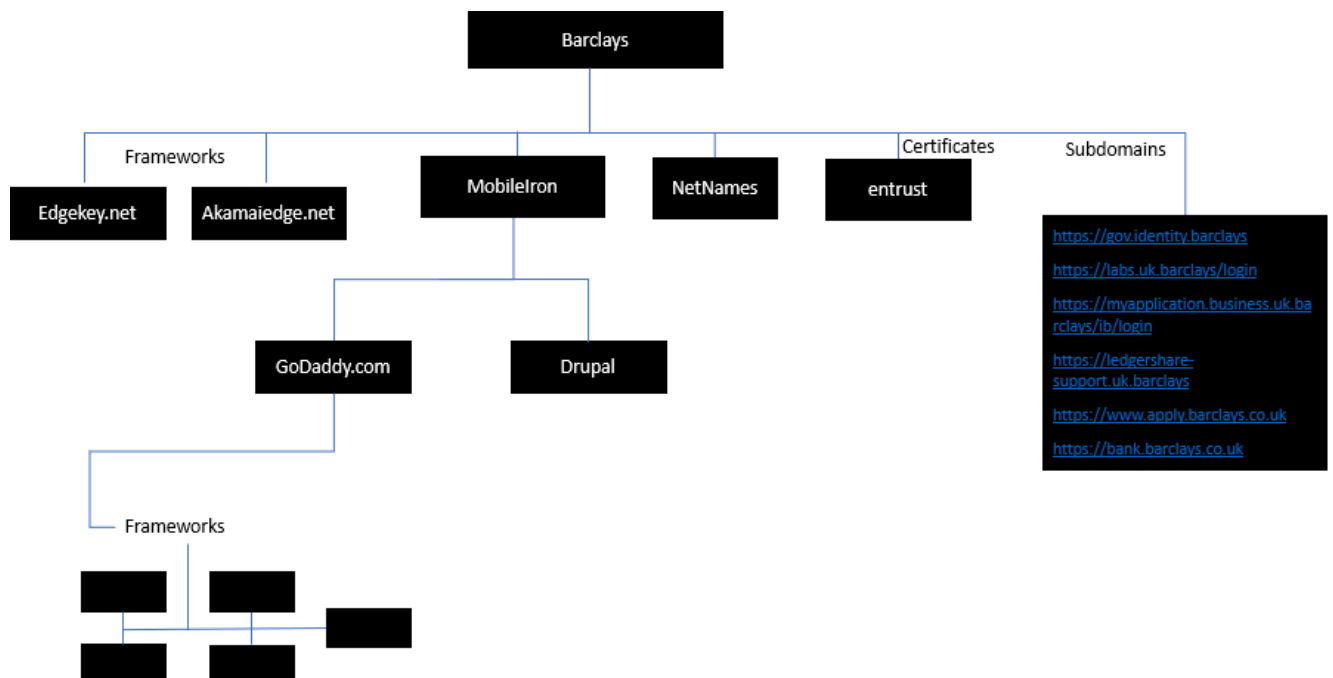
Entrust is a certificate issuer thus could allow attackers to issue their own certificates as well as revoke real ones, this could be a security issue. Attackers could also generate their own certificates to impersonate their web services to compromise the target organizations domain. We could try and see who issues the certificates for entrust.

	Criteria		Type: Identity	Match: ILIKE	Search: 'www.entrustdatacard.com'	

crt.sh ID	Logged At ⇧	Not Before	Not After	Matching Identities	Issuer Name
:192506758	2020-08-04	2020-08-04	2021-08-04	www.entrustdatacard.com	C=US, O="Entrust, Inc.", OU=See www.entrust.net/legal-terms, OU="(c) 2 CN=Entrust Certification Authority - L1M
:053322882	2019-10-30	2019-10-28	2022-01-27	www.entrustdatacard.com	C=US, O="Entrust, Inc.", OU=See www.entrust.net/legal-terms, OU="(c) 2 CN=Entrust Certification Authority - L1M

What a surprise. They issue their own certificates, no issues there apart from if attackers compromise their type of certificates they could exploit the Barclays domain as well as the entrust domain. No third parties are being shown from the certificate authorities for entrust.

Using Google Dorks on entrust

Searching through Google with 'site:entruct.com' you can find only one subdomain which is:

https://api.managed.entrust.com

This API can be added to our supply chain diagram.
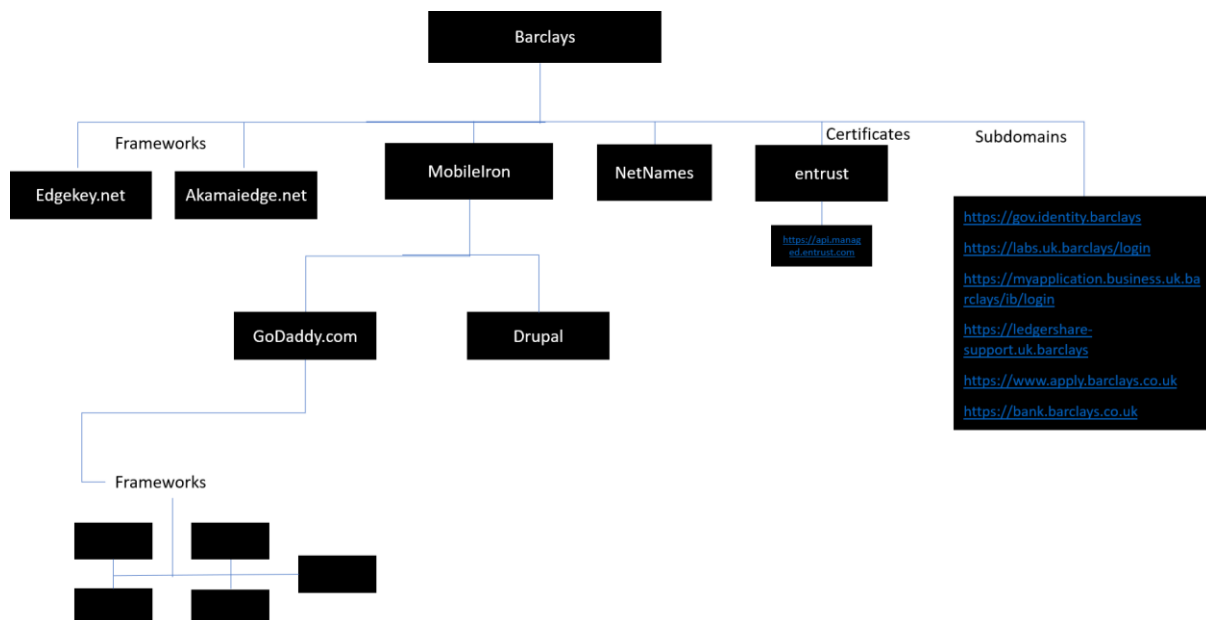
Let us move onto a third party of Mobile Iron, GoDaddy.com, the hosting provider of Mobile Iron, which implements security for Barclays.co.uk.

The GoDaddy Third Party

www.godaddy.com

GoDaddy is an American publicly traded Internet domain registrar and web hosting company, they provide services to businesses that wish to buy a domain and host it on their web servers. Mobile Iron uses GoDaddy for either their domain name or for web hosting.

Whois

Running a 'whois' on GoDaddy returns a list of their named servers, but that is about it, this could be a potential attack vector if they outsourced their servers, as it could identify third parties.

Crt.sh - GoDaddy Certificates

Using crt.sh returns that GoDaddy issues their own certificates, no third parties can be found here.

Google Dork searching

Cameron Noakes

site:*.godaddy.com

🔍 All   🖼 Images   📰 News   🏷 Shopping   📍 Maps   ⋮ More

This Google Dork query returns some of the GoDaddy subdomains, most of them are country code subdomains to ensure the customers can read the content on their website (e.g uk.godaddy.com, ru.godaddy.com etc), I found some that aren't country coded:

https://careers.godaddy.com/

http://auctions.godaddy.com/

These were the only two subdomains I could enumerate, GoDaddy does a very good job at hiding the subdomains.

HTTP Subdomain Issue

One thing that stood out about these two subdomains is that one is using HTTP, which sends un-encrypted data across the internet, this is the auction website.

Popular Searches		Search Results to Return	
Most Active	⌄	25	⌄

Featured Domains						View all Featured Domains ⊠
✔ **Name**	**Bids/Offers**	**Traffic**	**Estimated Value**	**Price**	**Enter Bid/Offer**	**Time Left**
☐ ⊕ snagaquote.com	0	-	-	£2,293 *	Buy Now for $3,000 *	41D 20H
☐ ⊕ backmyloan.com	0	-	-	Make Offer	USD$ Offer $10,000 or more	15D 20H
☐ ⊕ theamericandream2020.com	0	-	-	Make Offer	USD$ Offer $8,000 or more	23D 20H
☐ ⊕ futbolclubbarcelona.com	0	-	-	£38,209 *	USD$ Bid $50,000 or more	3D 14H

This subdomain handles bidding and payment for domains, this could be a potential GoDaddy vulnerability due to attackers being allowed to intercept the private data transferred across the internet such as sensitive banking information. However, once decided it then redirects the user to a secured Godaddy.com website to buy the domain securely.

Time to add these subdomains to our supply chain diagram and take out the GoDaddy framework placeholders.

Cameron Noakes
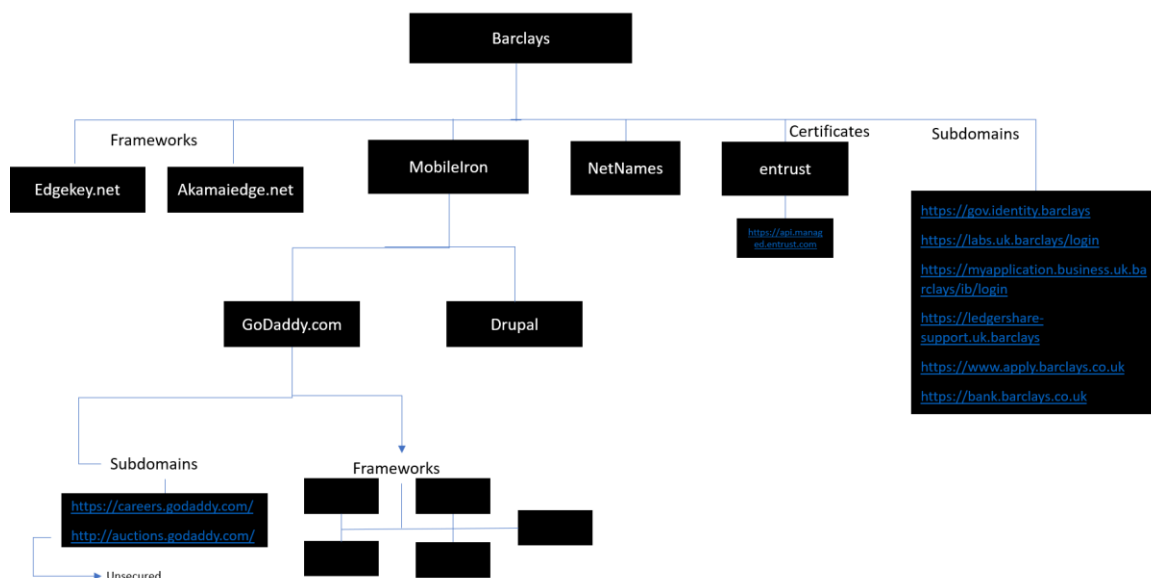
Whois

Running a whois and a dig query on Mobile Iron returns some information that was overlooked the first time, the IP range of 23.185.0.1 -> 23.185.0.255 (as this is the highest range it could be).

Plugging 23.185.0.1 into a browser returns a 404 error on a site called 'Pantheon.io', another useful third party that could be of great use.

Looking into the Pantheon.io website I managed to find a similar amount on robots.txt as I did with Mobile Iron, there is 'MAINTAINERS.txt' which referenced 'Drupal', the same company that did software management for Mobile Iron. Since this is now linked directly to a third party of Mobile Iron we can choose two locations to insert Pantheon

Cameron Noakes

into the supply chain diagram, either after Mobile Iron or after Drupal since they are both linked, due to the structure of my supply chain diagram and how I found this DNS A record of Mobile Iron that was actually from Pantheon, it is fair to put it after Mobile Iron.
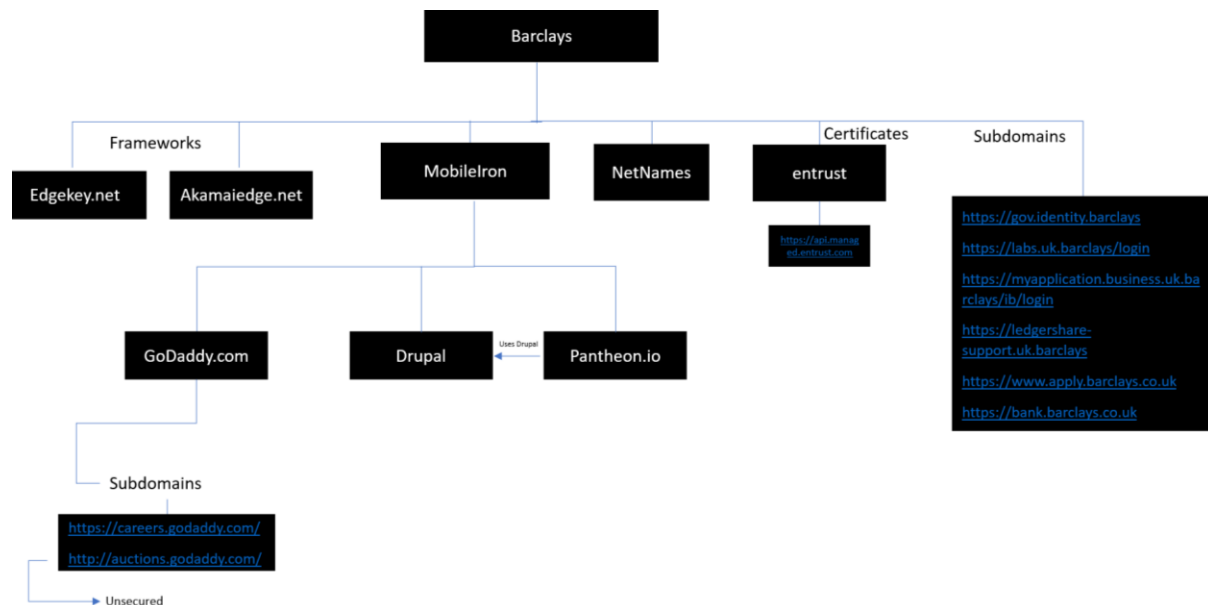
DETECTION SYSTEMS FOR THE SUPPLY CHAIN

Now that we have successfully mapped Barclays, we can start to think about detection systems on what we would like to monitor. It was key to have a picture of what Barclays supply chain relies on to make sure adequate and accurate detection systems could be thought of and put in place.

For this detection system we are going to be targeting the IP range of Pantehon.io as it will be our main focus due to it being on par with Mobile Iron.

DNS Record Monitoring tool

In order for me to code a DNS monitoring system I first need to get familiar with how one would go about this, I used 'www.pagerduty.com' to help me understand more about incident response and DNS records

PagerDuty is an American cloud computing company that produces a SaaS incident response platform for IT departments. Using PagerDuty, it helps to get my head around some of this as I have not had much experience with DNS records, automated tooling,

Cameron Noakes

and incident response. They have a free two weeks demo so signed up for it to look around.

Description of the DNS tool I will create

I will use Python to code and hopefully use any DNS libraries that could be of use, I have the IP ranges of Mobile Iron from previous reconnaissance mentioned above, we will compare the Answer (A) DNS records to that of the IP range of Mobile Iron, if the DNS A record lies outside the scope of the IP range then it will flag it up, for example an IP in the range changes to one in china would be a bit unusual.
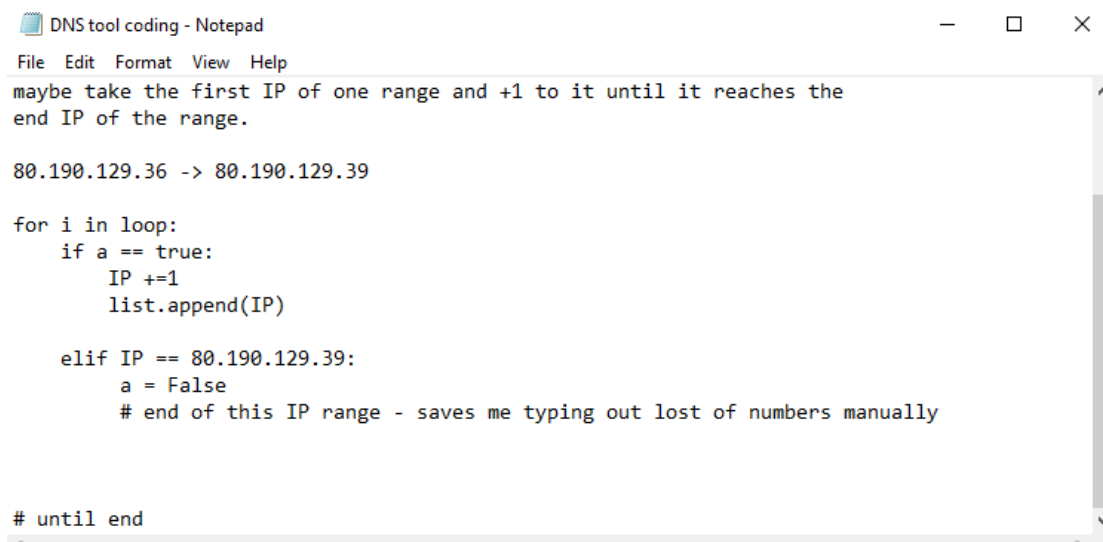
I used a library called 'dnspython', which is a DNS lookup tool that helps to provide a similar output to that of dig in Linux. From this I could run queries of data directly from the built-in functionality of the library.

I'm looking for 'A records' so our query will be specific for that. I then had to use 'response.answer' in order to fetch the answer I wanted, which was the 'A record', and then convert the output to text otherwise it would give me the storage location of it.

```
25  # A records
26  import dns.resolver
27  A = dns.resolver.query("mobileiron.com", 'A')
28  for i in A.response.answer:
29      for j in i.items:
30          print (j.to_text())
31
```

Cameron Noakes

This returned the 'A record' DNS response of: 23.185.0.1. This IP looked too generic, like a default gateway so I had some uncertainty, so I ran a quick dig query on Mobile Iron to check and it is correct. We now have to setup a successful DNS Answer (A) record query program and make sure it is accurate.

Now we want to make the program to detect any changes in the A record. I then had another look at the Ripe full text search again for the IP ranges. I wanted to see how I could use the IP range without becoming inefficient and manually typing out all the IPs in each range. After a little, while I put together some python code as a template to see how it would work in my head, a sort of visual representation for my coding brain, here it was:

```
DNS tool coding - Notepad                                    —    □    ×
File  Edit  Format  View  Help
maybe take the first IP of one range and +1 to it until it reaches the
end IP of the range.

80.190.129.36 -> 80.190.129.39

for i in loop:
    if a == true:
        IP +=1
        list.append(IP)

    elif IP == 80.190.129.39:
        a = False
        # end of this IP range - saves me typing out lost of numbers manually


# until end
```
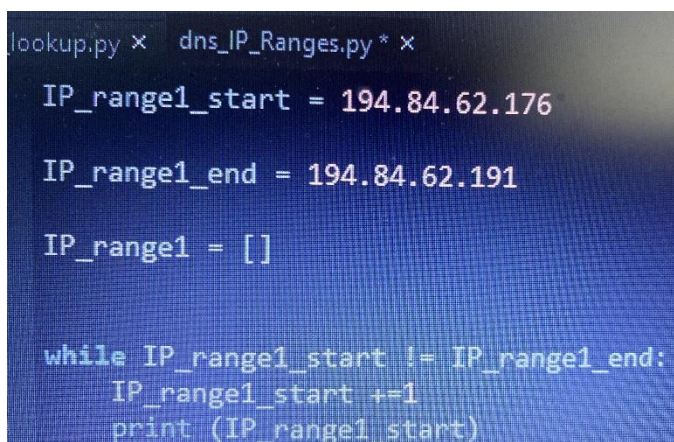
This was just to help get my head around how I would implement it and how it could possibly look. This is not the finished code but just a plan.

I then started trying to put it together in an IDE, I changed a few things upon development the original first development is seen below.
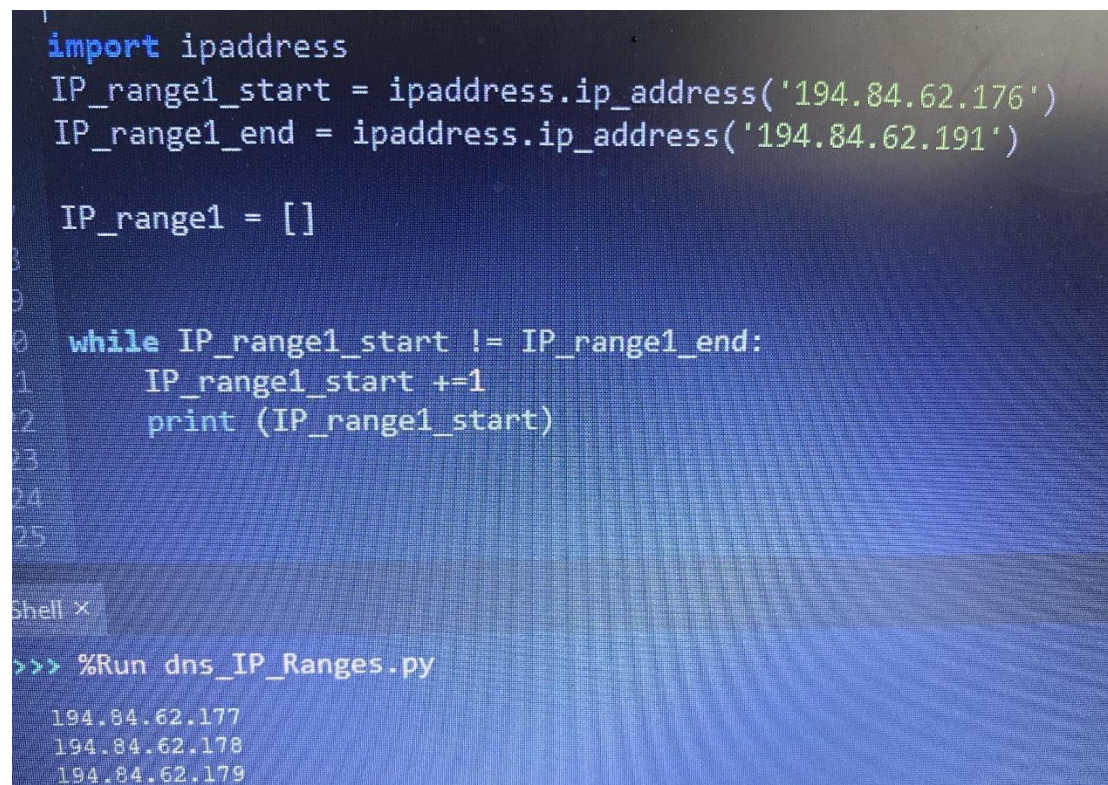
```
lookup.py ×    dns_IP_Ranges.py * ×
 IP_range1_start = 194.84.62.176

 IP_range1_end = 194.84.62.191

 IP_range1 = []



 while IP_range1_start != IP_range1_end:
      IP_range1_start +=1
      print (IP_range1_start)
```

Cameron Noakes

When I went to run it did not recognise the IP addresses as a valid data type which meant it could not be used in a loop to generate the list of IP ranges. I would also have an issue with adding one onto them due to the decimal points messing things up.

I came across a built-in Python library that allowed the use of IP addresses, it allowed them to be used as integer numbers (even though they had multiple decimal points). I then adapted my code to fit this library and was returned the IP address range of Mobile Iron, the first 'inetnum IP range' from Ripe full text search, this was just some dry data. This was a more efficient method of getting the IP ranges inside Python without having to type several numbers inside of a list.

```python
import ipaddress
IP_range1_start = ipaddress.ip_address('194.84.62.176')
IP_range1_end = ipaddress.ip_address('194.84.62.191')

IP_range1 = []


while IP_range1_start != IP_range1_end:
    IP_range1_start +=1
    print (IP_range1_start)
```

```
Shell ×

>>> %Run dns_IP_Ranges.py

  194.84.62.177
  194.84.62.178
  194.84.62.179
```

I can now save this IP range to a list and amend the IP range to that of the DNS A record range.

I ran a whois on 23.185.0.1, which is the DNS A record result that was returned, and the IP range used is 23.185.0.1 -> 23.185.0.255. We can change to this IP range after we have the main code working, for now we will stick with the '194.84.62.XXX' range.

I amended the generation of the range to the list for the *IP_range1* variable to allow the contents to be stored which will allow the comparison later on.

```
import ipaddress

IP_range1_start = ipaddress.ip_address('194.84.62.176')
IP_range1_end = ipaddress.ip_address('194.84.62.191')

IP_range1 = []


while IP_range1_start != IP_range1_end:
    IP_range1_start +=1
    print (IP_range1_start)
    IP_range1.append(IP_range1_start)
    print (IP_range1)
```

Shell ×

```
Pv4Address('194.84.62.182'), IPv4Address('194.84.62.183'), IPv4Address('194.84.62.184'), IPv4Address('1
194.84.62.187
[IPv4Address('194.84.62.177'), IPv4Address('194.84.62.178'), IPv4Address('194.84.62.179'), IPv4Address(
Pv4Address('194.84.62.182'), IPv4Address('194.84.62.183'), IPv4Address('194.84.62.184'), IPv4Address('1
4Address('194.84.62.187')]
194.84.62.188
```

This was the top section of the coded DNS tooling system; it holds the start and end IP addresses in the range to then calculates the IP addresses in between for the range.

The IP range was also added in.

dns_IP_Ranges.py ×

```
import dns.resolver
import ipaddress

IP_range1_start = ipaddress.ip_address('23.185.0.0')
IP_range1_end = ipaddress.ip_address('23.185.0.255')


IP_range1 = []

while IP_range1_start != IP_range1_end:
    IP_range1_start +=1
    IP_range1.append(IP_range1_start)


# A records
A = dns.resolver.resolve("mobileiron.com", 'A')
for i in A.response.answer:
    for j in i.items:
        A_record = j.to_text()
        print ("IP Answer (A) record:",A_record)



for item in IP_range1:
    if A_record == item:
        print ("The DNS A record for the domain is in the IP address range. ")
        break
```
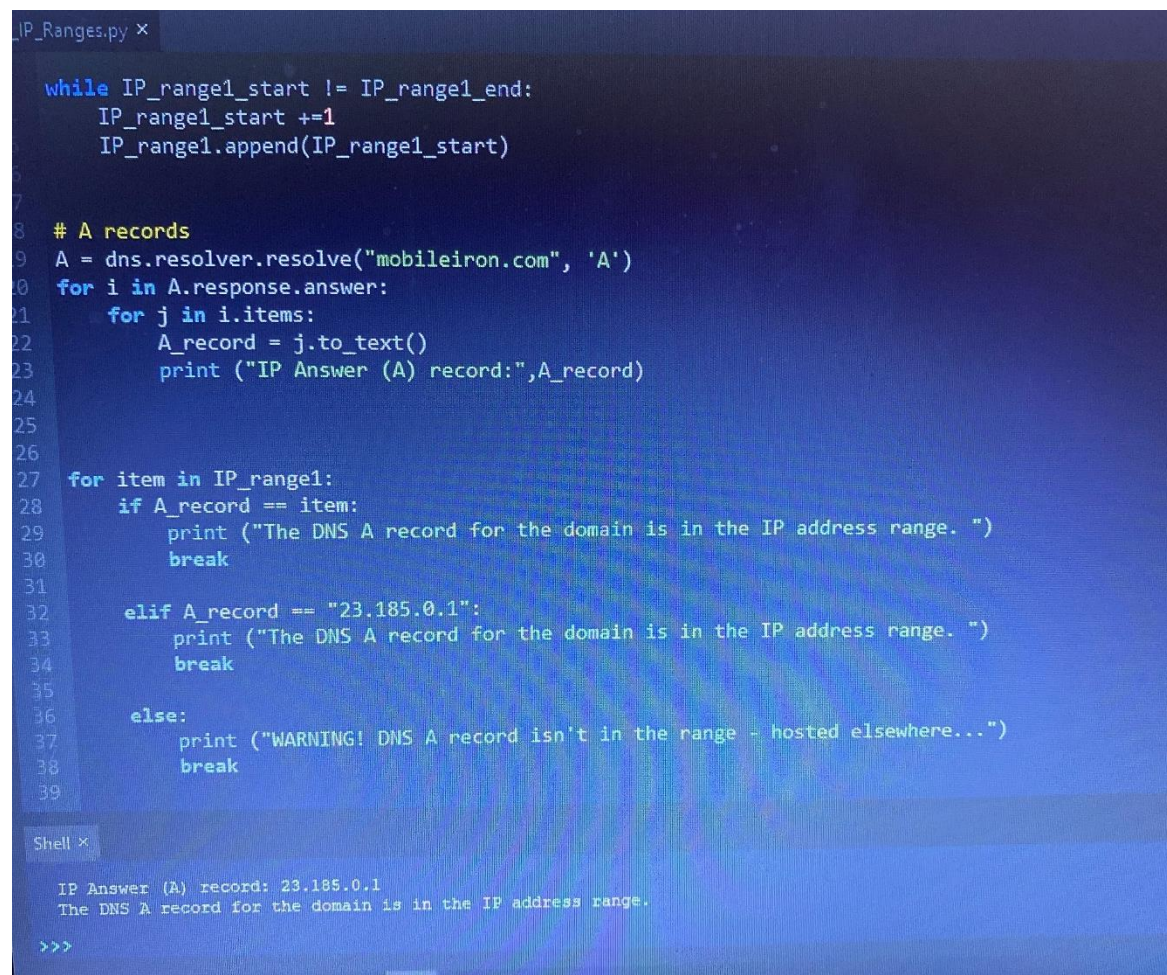
Shell ×

```
IP Answer (A) record: 23.185.0.1
The DNS A record for the domain is in the IP address range.
```

Cameron Noakes

The bottom half of my code is for the linear search comparison, Python was being a little buggy when testing so had to add an 'elif' in as it was not registering the first IP in the range which happened to be the current DNS A record IP of 23.185.0.1.

```python
IP_Ranges.py ×

    while IP_range1_start != IP_range1_end:
        IP_range1_start +=1
        IP_range1.append(IP_range1_start)


    # A records
    A = dns.resolver.resolve("mobileiron.com", 'A')
    for i in A.response.answer:
        for j in i.items:
            A_record = j.to_text()
            print ("IP Answer (A) record:",A_record)




    for item in IP_range1:
        if A_record == item:
            print ("The DNS A record for the domain is in the IP address range. ")
            break

        elif A_record == "23.185.0.1":
            print ("The DNS A record for the domain is in the IP address range. ")
            break

        else:
            print ("WARNING! DNS A record isn't in the range - hosted elsewhere...")
            break

Shell ×

    IP Answer (A) record: 23.185.0.1
    The DNS A record for the domain is in the IP address range.

>>>
```

The above screen dump is the DNS tooling concatenation with the IP range generator, together they generate all IP addresses in the range and will perform a linear search through them all for the DNS A record. The program will then return a dynamic response depending on whether or not the DNS A record is location inside of the IP range, if it is not then it will flag up that it sits outside of the IP range, which could become a security issue.

How will I host this tool?

AWS lambda will be used to host the code and continuously schedule a runtime for it, so it checks the DNS A record automatically and creates an alert system.

Cameron Noakes

AWS Lambda

I setup an AWS Lambda account, costed 79p to verify my card and account. Once I had the account I went over to the Lambda section of AWS and created a new function as well as learnt more about AWS as I have not used it before.

I pasted my code into a new file and called it 'DNS_tooling.py', I then contacted my friend Kaspar to help with how to get my code to work due to the DNS library not being built into Python.

I then used pip to install the dnspython library again but in the same folder as my code, I renamed my code file to 'mycode.py', i used the pip command to reinstall in a different location:

Pip install dnspython -t ./

This installed it in my current directory which was where mycode.py was, compressed it and uploaded to AWS lambda, I then added the handler function to allow my code to work correctly and changed the handler directory in 'basic settings' further down the page so it links correctly.

After deleting functions and restarting a few times I managed to get my code fully working with the modules it needed to run.

Alert system

I then wanted to create some sort of alert system to alert me when it sits outside the range, I researched 'Amazon Simple Notification Service' (AWS SNS) to allow alerts, I wanted an alert to be sent to my phone, I wrote the code needed and understood it to then only realise that there was a publicly available AWS SNS list for sending messages to countries, and England was not on the list. England is not supported in receiving SMS message alerts, so will have to try email instead.

I had to change around some variables due to it not being for SMS text messages anymore e.g swapping PhoneNumber to TopicArn, AWS has placeholders which need to be filled in order for it to work but some are specific to emails and some are specific to other types of alerts like SMS texts.

I copied my ARN topic string that was used to add my email to AWS through SNS, I entered this into the TopicArn variable as well as added a message and setup boto3.client and client.publish code.

Cameron Noakes

After debugging my code for 10 minutes wondering why on earth it was not working, I realised I forgot a comma on one of the lines.

For test purposes I swapped out 'j.text()' in my Python code which was the DNS A record to that of 23.185.1.0 which was purposely out of range, I was then sent an email about it being outside of the range.

From my detection system using AWS SNS and my lambda code, I now get emails about if the DNS A record is sitting out of range.
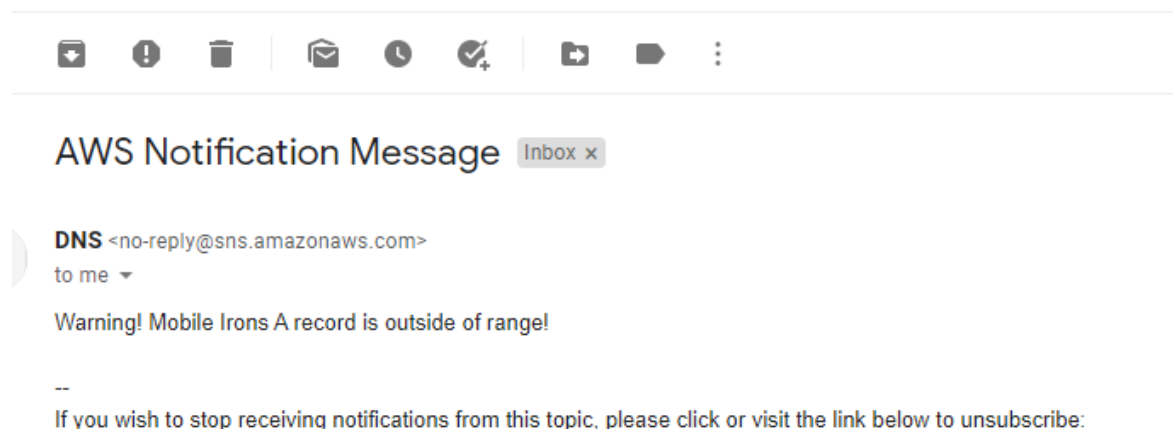
Here is the complete AWS SNS and Python code for this detection system:

Cameron Noakes

```
1   import dns.resolver
2   import ipaddress
3   import boto3
4
5   def handler(a,b):
6
7       client = boto3.client(
8           "sns",
9           aws_access_key_id = "████████████",
10          aws_secret_access_key = "████████████████████████",
11          region_name = "us-east-2"
12          )
13
14      client.publish(
15          TopicArn = "arn:aws:sns:us-east-2:686688756380:dns_tooling",
16          Message = "Warning! Mobile Irons A record is outside of range!"
17          )
18
19      IP_range1_start = ipaddress.ip_address('23.185.0.0')
20      IP_range1_end = ipaddress.ip_address('23.185.0.255')
21
22
23      IP_range1 = []
24
25      while IP_range1_start != IP_range1_end:
26          IP_range1_start +=1
27          IP_range1.append(IP_range1_start)
28
29
30      # A records
31      A = dns.resolver.resolve("mobileiron.com", 'A')
32      for i in A.response.answer:
33          for j in i.items:
34              A_record = ipaddress.ip_address('23.185.1.1') #j.to_text()
35              print ("IP Answer (A) record:",A_record)
36
37
38
39      for item in IP_range1:
40          if A_record == item:
41              print ("The DNS A record for the domain is in the IP address range. ")
42              break
43
44          elif A_record == "23.185.0.1":
45              print ("The DNS A record for the domain is in the IP address range. ")
46              break
47
48          else:
49              print ("WARNING! DNS A record isn't in the range - hosted elsewhere...")
50              break
51
```

Line 34 would need to be adjusted to take out the test IP and uncomment the j.text()

Automation

I then went into CloudWatch and clicked on rules and scheduled it to run every hour to make sure that if there were an issue it would only take maximum 1 hour to identify it and start to resolve the issue.

Rules > DNS_schedule

Summary

ARN   arn:aws:events:us-east-2:686688756380:rule/DNS_schedule

Schedule   Fixed rate of 1 hours

Status   Enabled

Description   runs every hour to check DNS A record is in IP range

Monitoring   Show metrics for the rule

Targets

Filter:

Type	Name	Input	R
Lambda function	DNS_tooling	Matched event	

Cameron Noakes

My AWS Lambda files and folders

```
IP_range1 = []

while IP_range1_start != IP_range1_end:
    IP_range1_start +=1
    IP_range1.append(IP_range1_start)


# A records
A = dns.resolver.resolve("mobileiron.com", 'A')
for i in A.response.answer:
    for j in i.items:
        A_record =  j.to_text() #to test recieve emails: ipaddress.ip_address('23.185.1.1')
        print ("IP Answer (A) record:",A_record)
```

Execution Result

▼ Execution results

```
Response:
null

Request ID:
"6a1d2939-1a13-4193-b4b8-46824b5a4577"

Function logs:
START RequestId: 6a1d2939-1a13-4193-b4b8-46824b5a4577 Version: $LATEST
IP Answer (A) record: 23.185.0.1
The DNS A record for the domain is in the IP address range.
END RequestId: 6a1d2939-1a13-4193-b4b8-46824b5a4577
REPORT RequestId: 6a1d2939-1a13-4193-b4b8-46824b5a4577  Duration: 210.16 ms Billed Duration: 300 ms Memory Size: 128 MB Max Memory Used: 74 MB
```

OTHER ORGANIZATION FIXES AND SOLUTIONS

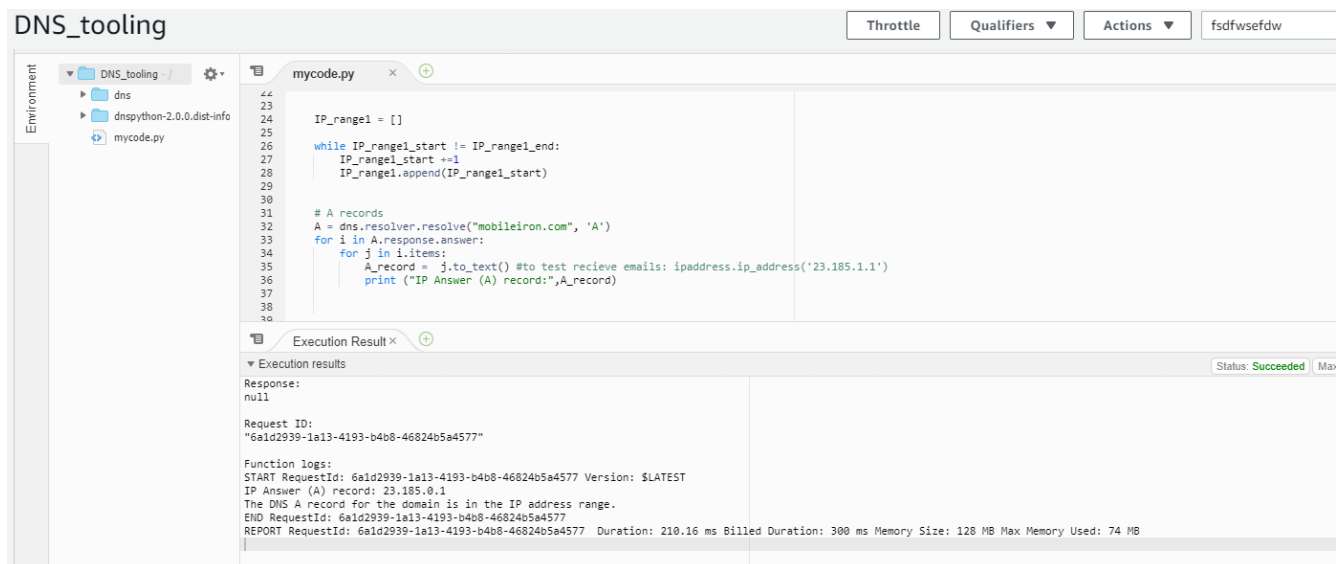Organizations need to implement secure security measures to prevent supply chain attacks on various supply chain components as well as make sure to restrict access to critical infrastructure to act as an access level that attackers would not be able to pass or bypass.

Organizations also need to configure firewalls fully to make sure the organizations networks are safe and secure and to filter any incoming and outgoing traffic.

Another security measure could be to put an interception proxy between the firewall and their organizational network to ensure any traffic gets logged and can be traced back if there was any malicious activity directed towards their networks.

As an organization you could also run training exercises on staff to make sure they are fully aware of any potential threats such as phishing emails and phone calls, this would prevent the potential situation of an attacker getting onto either the network or server or retrieving company and client sensitive data.

Developers of the organization can implement incident response software and detection systems to ensure attackers are found early on.

Cameron Noakes

Attacks against organizations are growing more over the years and are targeted more so against an organizations software as it can potentially be exploited due to developers not implementing the right amount of security practices. As an organization you should implement some sort of security training course for developers to make them aware of pressing security issues with websites and software.

SUMMARY/ WHAT I HAVE LEARNT

Overall I have learnt immense amounts about supply chains, supply chain attacks, OSINT, external perimeter mapping, subdomains, third parties, links between companies, passive reconnaissance, development of tooling, development of detection systems, AWS, AWS lambda and CloudWatch as well as using Python libraries that I have not used before.

I have really enjoyed exploring this blog/ research project from literally no knowledge of what a supply chain attack even was, I was persistent to the days where I found very little OSINT and was rewarded by having days of constant focus and productivity, I now have very in-depth knowledge of supply chain attacks and development of detection systems.

Originally I was overwhelmed about how I would implement security detection tools to run continuously and be connected to the internet, now I understand how someone would go about implementing detection systems.

Now that I know how to work AWS and Lambda specifically, I can use this platform for new and upcoming projects that I wish to automate or ones that need to continuously run.

I did not realise how important open source intelligence (OSINT) is for pen tests and from the point of view of an attacker, now that I understand them to a good degree I should hopefully be able to secure systems from the knowledge I have obtained from completing this research project/blog.

RESOURCES USED

Cameron Noakes

https://www.fireeye.com/blog/threat-research/2020/03/monitoring-ics-cyber-operation-tools-and-software-exploit-modules.html

https://www.youtube.com/watch?v=uXm2XNSavwo

https://www.youtube.com/watch?v=vno8xm--K44&t=426s

https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/

https://appsecco.com/books/subdomain-enumeration/active_techniques/brute_force_dictionary.html

https://www.youtube.com/watch?v=Mi1QBxVjZAw

https://www.thebci.org/news/what-are-the-effects-of-reputational-damage.html

https://blog.checkpoint.com/2017/09/14/expensivewall-dangerous-packed-malware-google-play-will-hit-wallet/

https://www.websecurity.digicert.com/security-topics/how-code-signing-works

https://en.wikipedia.org/wiki/MobileIron

https://apps.db.ripe.net/db-web-ui/fulltextsearch

https://www.google.com/search?q=AWS+lambda+sns+An+error+occurred+when+calling+the+Publish+operation%3A+Invalid+parameternot+valid+to+publish+to%22%2C&oq=aws&aqs=chrome.0.69i59l3j69i57j35i39j0j69i60l2.406j0j7&sourceid=chrome&ie=UTF-8

https://us-east-2.console.aws.amazon.com/

https://www.youtube.com/watch?v=j6OwZsS6AmU

https://codebeautify.org/htmlviewer/

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sns.html

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sns.html#SNS.Client.get_sms_attributes

https://www.mobileiron.com/en

https://uk.godaddy.com/

https://www.youtube.com/watch?v=j6OwZsS6AmU&t=2s

https://forums.aws.amazon.com/thread.jspa?threadID=287110

https://github.com/pypa/pip/issues/5093

https://www.google.com/search?q=comparing+DNS+A+record+to+IP+ranges&oq=comparing+DNS+A+record+to+IP+ranges&aqs=chrome..69i57.11431j1j7&sourceid=chrome&ie=UTF-8

Cameron Noakes

Cameron Noakes