KTH Royal Institute of Technology
Stockholm

School of Electrical Engineering and
Computer Science

Building Networked Systems Security EP2520

System design and implementation details for ACME Network Project

Authors
Thomas Poncet *(trfpo@kth.se)*
Alexios Stournaras *(alexioss@kth.se)*
Emil Ståhl *(emilstah@kth.se)*
Andreas Åström *(aastro@kth.se)*

March 2022

Contents

1 ACME needs and requirements

In this report, we share our system design, analysis, and implementation of a enterprise network infrastructure as requested by ACME Scandinavia, headquartered in Stockholm. ACME wishes to securely extend their headquarters IT environment to their new branch office in London and enhance their employees access to the company IT resources. ACME needs a new network infrastructure including a new secure web-server in the headquarters of Stockholm. Users at each branch should be able to reach resources located at the other network. Moreover, employees need to be able to connect to the corporate Wi-Fi with their laptops and smartphones as well as share files between each other. If employees are not using their cryptographic credentials they need to use their trusted device for Two-Factor Authentication (2FA).

2 System Design

This section describes the proposed implementation and chosen technologies to achieve the needs specified by ACME.

2.1 Implementation Design

The proposed network topology and design for achieving ACME's needs are illustrated in Figure 1. The figure shows the ACME Stockholm headquarter at the top and the ACME London branch to the bottom right. The Stockholm branch runs one server that hosts all of the required services including FreeIPA, Nextcloud, IDS, FreeRadius, and the VPN Server. The two sites are connected through an OpenVPN tunnel. At the bottom left the figure illustrates a client connecting from an offsite location using 2FA and OpenVPN.

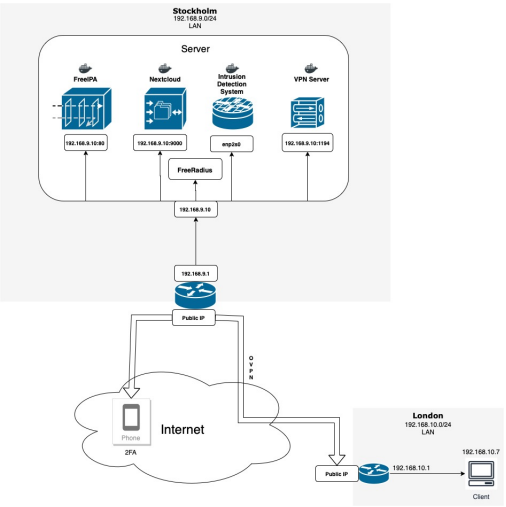

Figure 1: Proposed topology of the new network for ACME

2.2 Security Requirements

This section presents ACME's security requirements and what technologies we have chosen to meet these requirements.

2.2.1 Employee Authentication

For employee authentication we use FreeIPA whih is an open source identity manager. FreeIPA allows for authenticating a user with 2FA and username/password and provides a centralised authority mechanism.

2.2.2 Confidentiality

Communication between the London and Stockholm branch goes through a VPN tunnel. Meaning that the information between the two ACME sites is hidden for third parties. Any outside connections to the Stockholm office are also going through a VPN tunnel, this is useful for employees using their laptop at home or other locations.

2.2.3 Secure connectivity

Accessing the file server located at the Stockholm office requires 2FA. The file server is only accessible for employees with network access, either through Stockholm, London branch, or connecting through the VPN tunnel remotely.

2.2.4 Secure Wireless Access

The tool chosen for authentication and approving network access for users was FreeRadius, an open source RADIUS server. It will handle communication with the routers. It will authorize certain username/password (employees) combinations and approve them for network access. Unfortunately, issues regarding the connection between FreeRadius, FreeIPA and the router emerged. Therefore, we manually update both for any changes and not just FreeIPA. The FreeRadius implementation was done with EAP-TLS using certificates, though self-signed. More details on that in section 3.

2.2.5 Secure File Exchange

NextCloud is the chosen file exchange tool. It connects to the FreeIPA database and only allows users with permission to access it. Users can easily exchange and share files using Nextcloud's capabilities.

2.2.6 Other Security

To be alerted about potential ongoing attacks against the network infrastructure, we are running an Intrusion Detection System (IDS) using the open source IDS software named Snort. The IDS software runs containerized on a server located in the Stockholm branch. The IDS is configured to prioritize alerts regarding outside threats, meaning traffic originating from outside the ACME networks that has a destination residing inside the ACME networks. Regarding inside threats Snort is configured to alert system admins regarding malicious activity such as multiple failed SSH authentications.

3 Discussion

In this section, we discuss our set-up and reflect on how the requirements, functionality, and the security of our solutions are satisfied considering specific use cases.

3.1 Routers

The routers have a simple DD-WRT set up running. After we configured the routers to run DD-WRT properly and give out IPs from the networks 192.168.9.0/24 and 192.168.10.0/24 for Stockholm and London respectively, we created an internal VPN connection between them. This was achieved by creating a new certificate authority, certificates and public and private keys from them with OpenSSL. Since this connection is always open and doesn't have anything to do with the users, we decided to not connect them with FreeIPA and have them set up like that independently. Moreover, we connected the routers with FreeRadius (discussed below), by making them clients of FreeRadius. As such, they can allow internet access via Wi-Fi with EAP-TLS encryption.

3.2 System cooperation for achieving security requirements

Our implementation consists of two routers and one host machine acting as a server for the docker containers running our different services. The two routers, one for each branch, hosts two interfaces, as shown in Figure 1. One of the interfaces for each branch is responsible for communication between the branch and the outside world, while the other is dedicated to handle communication with the respective interface located at the other branch. For the purposes of the demonstration, we haven not created a VPN tunnel from the Internet to London, but by following the same steps as described in the appendix, it is trivial to do so. Figure 1 demonstrates the proper set up, but we will demonstrate only the Stockholm branch to be more concise. The tools that are required for the routers is OpenVPN and DD-WRT. OpenVPN is used to ensure encrypted file exchanges and communication between the branch and the headquarters gateways. We create one VPN connection for London and Stockholm, where the Stockholm router will act as an OpenVPN server, and one VPN connection from an off-site locations to the Stockholm branch, with the router-gateway of Stockholm acting as the server. Thus, we need two public IPs for this demonstration.

DD-WRT allows for greater router configurability and can also act as a firewall. To be more precise, as the reviews suggested, we have firewalls in each of the branches that prevent access to the internal networks, unless the traffic comes from the VPNs that we deploy.

3.2.1 FreeIPA

The host machine is responsible for running an Authentication Server (AS), File Server (FS), IDS and an OpenVPN Access Server on it as containers, FreeRadius runs normally without a container. To act as an AS, the host machine runs FreeIPA, an open source identity management system. FreeIPA takes advantage of Kerberos, LDAP, bind and Dogcert, which is responsible for handling CAs and certificates, and utilises them to become a centralised authority. When a new user is created, it is added to the LDAP and in order to access LDAP safely FreeIPA uses its CA and its certificates. As such, whenever we add another host to the "FreeIPA network" and in its kerberos domain, they get a certificate from that authority that allows them to safely access LDAP and verify the user that tries to log in to their services. Moreover, we do not have to create a user in every different service that we use (i.e Nextcloud, Openvpn access server etc) but we only use the FreeIPA users, which we can easily add and manage either for the WEBUI that it offers or from the command line, and we connect the services with it via LDAP. This allows each employee to have their identity verified by digital certificates, published by a trusted CA and also allows for Two-factor authentication (2FA) for employees, if we choose to. In our case, we do not really add hosts in the "FreeIPA network" because everything runs in containers on the same host machine. But through the docker container networking, the same principles apply and we can authenticate everyone through FreeIPA.

3.2.2 Nextcloud and FreeRadius

As for Nextcloud, we run a simple version of it in a container, and by integrating it with FreeIPA through LDAP, FreeIPA users can share their files through Nextcloud with one another. In theory, the same procedure is followed by connecting FreeRadius with FreeIPA, and having the router, connected via ethernet with the host machine, to verify users trying to access the Wi-Fi with Freeradius. However, while we managed to connect Freeradius with FreeIPA, the configurations of the router do not share the same encryption methods as FreeIPA's LDAP, so FreeRadius is not able to verify the users after all. As such, whenever we add a user in FreeIPA, we manually add them in the users file of FreeRadius as well, and then the router can properly verify these users by using WPA2 and EAP with TLS encryption and sharing a key with FreeRadius.

3.2.3 OpenVPN

Lastly, the host also runs an OpenVPN access server, to allow connections from the outside world to the private network. We realised that, although connecting the two branches with OpenVPN through the routers was running smoothly, using DDWRT and OpenVPN for multiple connections from outside was not optimal. As such, we handle the outside connections with the OpenVPN Access Server (OpenVPN-AS) running in a container on the host. We only need to allow port forwarding in our router to let that happen. Afterwards, we again connect OPenVPN-AS with FreeIPA through LDAP. When a new user is hired, he will also log in to OpenVPN as a client and download a file that allows him to connect through any device, like a mobile phone, that has an OpenVPN client application installed, with his username and password.

3.2.4 Intrusion Detection System

The requirements regarding intrusion detection is clearly satisfied thanks to how we have configured the IDS to function. In general, we consider the ACME LAN's to be relatively safe, meaning that any traffic flowing inside the ACME infrastructure is seen as legitimate. We consider types of attackers differently depending on their likely-hood. Since ACME has dedicated offices extensive security to enter the office buildings we consider situations where the attackers resides from within the ACME network to be relatively unlikely. On the contrary, we argue that attacks coming from outside ACME targeting open ports and public IPs to be considerably higher. Due to this, we have deployed tailored local IDS rules to detect for example ICMP request that has a source originating from outside ACME but a destination on our local LANs. However, we decided to monitor some traffic independently depending on where it comes from. As an example, the IDS catch situations involving a machine that is adversarial and keeps sending network discovery requests and probing the SSH connection of the machines for which it gets the IPs. This particular attack can be simulated with a simple nmap port scanning command and our IDS is proven to successfully detect these types of attacks. Other situations our IDS is configured

to detect are attacks regarding DDOS, NTP, an attack-responses. Lastly, we configured the IDS to alert about Log4j-attacks. Since we run all of our services on a single server, we decided to not configure port mirroring on the router for LAN-wide monitoring of attacks. However, this is not needed since all important services are running on containerized on the server which is analyzed by the IDS. However, it would be really easy to extend the infrastructure with port mirroring. The configuration of Snort would not need to updated since it already monitors the Stockholm network. The only thing to configure is the port mirroring itself, that is forward all traffic from the Stockholm gateway to the server where Snort is running. We consider this network to be just as safe with or without port mirroring, however if the network would to be extended with multiple servers then port mirroring would increase the security. Furthermore, the London branch currently have not an IDS either due to the same reasons as mentioned above, that is since the London branch has no services running in the network. However, an IDS could very easily be added to the London branch simply by doing a docker pull and run as described in the IDS section of the Appendix.

3.2.5 Some final details

It is important to note that due to the fact that we configured FreeRadius as described above, we decided to use 2FA everywhere, meaning both when someone connects with a VPN from outside as well as when we are inside the network and want to access a service like Nextcloud. Moreover, although normally we would have another similar set up in the London branch, for the purposes of this report we consider that users in London will connect to their Wi-Fi with authentication from FreeRadius in Stockholm, which is perfectly possible since we configured the internal VPN and we added the router of London as a client of FreeRadius.

Appendices

Appendix A

README - How to configure the network

A detailed description of how to set up the network is available at GitHub.