Team ISEAGE

Iowa State University

(Team 9)

White Team Documentation

Table of Contents

White Team Documentation

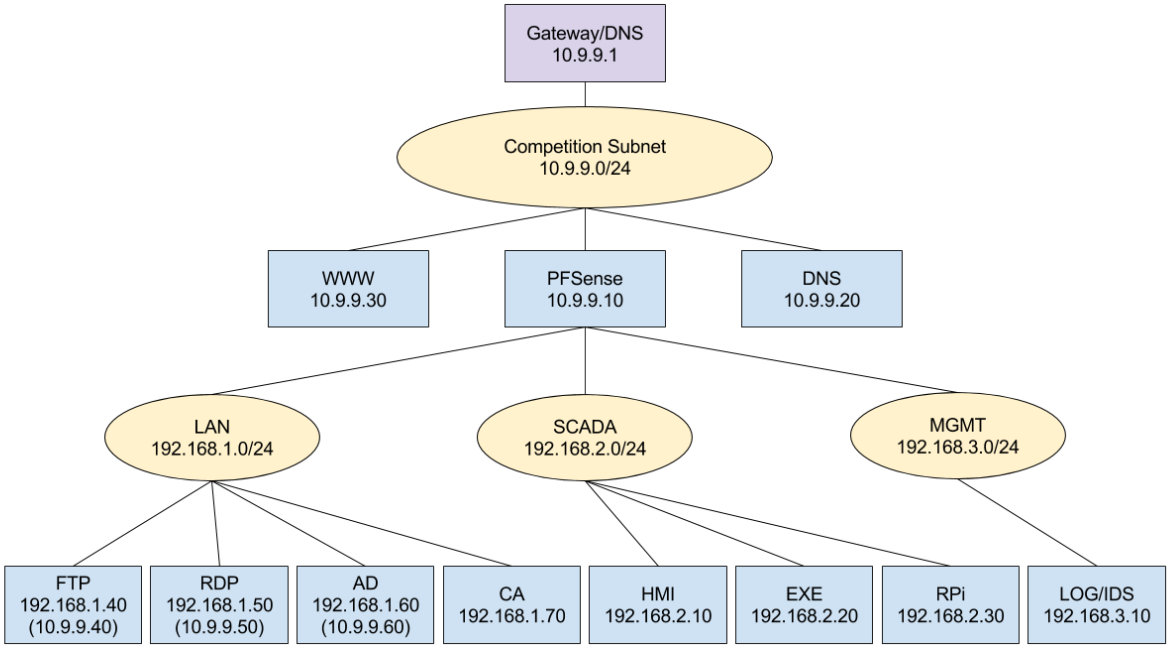
Public Services(Setup for Service Scanner)

We have the following services publically available:

Domain Name	IP	Port
www.site9.cchc.competition	10.9.9.30	22,80,443,10000(SSH,HTTP,HTTPS,Webmin)
ftp.site9.cchc.competition	10.9.9.40	22(SFTP/SSH)
rdp.site9.cchc.competition	10.9.9.50	3389(RDP)

Network Architecture

For our network setup, we decided to place some of our resources on the competition network and other resources behind a PFSense firewall. We allowed access to resources behind the firewall by utilizing virtual IPs and port forwarding. This allowed each service that required its own public IP, such as FTP and RDP, to be reachable from different IPs outside the network but still had that traffic pass through the firewall.



We segmented our internal network into three main networks that are under the control of PFSense. Via PFSense firewall rules tied to each network, we can control what traffic is allowed to flow between our three networks and the competition network. Rules in PFSense that are set on a certain interface/network are applied to traffic that *originate* from that interface/network. Its should also be noted that as with any network-based firewall, PFSense cannot block intra-subnet communications. In places we wanted to block certain intra-subnet traffic we utilized host-based firewalls such as iptables or Windows firewall.

LAN Rules

The LAN network contains the FTP, RDP, AD, and CA servers. With the exception of the RDP server being allowed to connect to the HMI Proxy system in the SCADA network, no systems in the LAN can connect to any of the systems in the SCADA network. Also none of the systems in the LAN can connect to the systems in the MGMT network. Systems in the LAN are allowed to make connections to systems in the competition network or the internet in general.

SCADA Rules

The SCADA network contains the HMI Proxy, EXE server, and Raspberry Pi PLC. The systems in the SCADA network are not allowed to connect to any other systems in any other networks period. If/when we want to perform system updates, the firewall rule preventing connections to the internet will momentarily disabled and reenabled once the updated have finished.

MGMT Rules

The MGMT network contains the Syslog/IDS server. This server does not need to connect to any other systems since PFSense makes the connection to the Syslog server to send it log files. Just as with the SCADA network, the systems in the MGMT network are not allowed to connect to any other systems in any other networks period. If/when we want to perform system updates, the firewall rule preventing connections to the internet will momentarily disabled and reenabled once the updated have finished.

Servers

Overall

OS hardening: The web, FTP, DNS, and internal management servers are all running a fresh install of Debian 8 Jessie. These systems are running fairly minimal software stacks and use a generalized configuration, distributed initially by Ansible.

IDS: All public-facing/production systems are running an OSSEC HIDS agent to collect security events at the host level.

Fail2ban: Public-facing Debian machines have fail2ban installed and configured to temporarily block IP addresses that have too many failed login attempts. This mitigates most attempts at bruce force logins.

Web Server

Outbound firewall rules: The server is prevented from opening new connections to untrusted servers.

Inbound firewall rules: Incoming connections are dropped unless they are on the allowed ports: 22, 80, 443, 10000.
System auditing: The system configuration was audited with the lynis audit tool.
MySQL: MySQL is bound to localhost only and *mysql_secure_installation* was run to remove any extra, unneeded options.
Nginx: Nginx is using a configuration only allowing modern, secure ciphers, and all web pages are served over HTTPS using a certificate provided by our CA. The server is using HSTS to prevent sslstrip attacks.

Application

The original web application was audited and found to be running a very insecure web application. The application, written in PHP, used some components from DVWA (an intentionally vulnerable application used for educational purposes), thus was very insecure and would be considerably difficult to fix.

To secure the application, it was redesigned from the ground up and written in Django, a modern Python web application framework. Django, when properly configured and used, has effective built-in protections against many common web vulnerabilities. The newly developed source code was audited by multiple team members.

- XSS: Django's templating engine performs proper HTML escaping to mitigate XSS attempts.
- CSRF: Django requires all forms to have valid CSRF tokens by default.
- Authentication: Django's built in authentication mechanisms handle all aspects of user authentication securely, using proper sessions and safety measures. It also allows for easy integration with our active directory environment.
- Authorization: Django makes it easy to implement strong function-level access controls that define who can access what.
- SQL injection: The Django ORM handles and abstracts all database operations while preventing SQL injection.
- File uploads/directory traversal: File uploads are handled by Django's backend and filenames are validated to prevent traversal.
- IDOR: ACLs and other authorization logic (X-Accel-Redirect passed internally to nginx) is in place to make files only downloadable by their owner.

Instead of using PHPmyadmin for user account management, a custom frontend was added the the Django application that allows full required management functionality while not requiring the full suite of tools provided by PHPmyadmin.

The webmin server was configured to use local authentication to only allow users with valid admin credentials to connect to it.

FTP Server

The FTP server runs off a standard installation of Debian. We've joined this server with our domain controller in order to compartmentalize the service with authentication/authorization. The FTP server is using SFTP to securely transfer files to and from the web server. The Employees are only allowed to access this server via ssh, however the Employees that are administrators are are allowed to access this server via SFTP and SSH. The Admins are allowed full access to the box to perform any maintenance that is required. When the Employees access this server via SFTP they are chroot'ed into a space that only they have access to. We have performed these these actions in order to make the system as easy to use as possible.

Domain Name Server

The DNS server runs a standard installation of Debian. We've joined this server with our domain controller. As a result, the Admins are the only users with access to the server over SSH; they can use this connection to manage the DNS records

Domain Controller

The Domain Controller runs on a standard Windows 2012 R2 server. We used the domain controller to centralize user management on the domain, This allowed us to easily perform security audits of the users and will allow us to quickly add or remove users from all systems, if needed. After a security audit, we determined that there were certain users that had been given excessive privileges on certain machines. Based on the documentation, we have decided that there are two accounts that should have administrative rights, cchcadmin and root, since most employees will not need that level of access unless their responsibilities cover that of a system administrator. If any other accounts are determined to need admin access, we will grant it at that point.

Remote Operations Server

The RDP service runs off of a standard Windows 2012 R2 server deployment. We've joined this server with our domain in order to compartmentalize the service with authentication/authorization. Users are classified based the needs for their role; as customers, employees, and administrators. Employees and administrators are given a

remote desktop to work with, access to the HMI, and use of FileZilla for SFTP. The HMI has an additional layer of authentication to ensure that someone leaving their remote desktop session open does not inherently mean someone else has access to the HMI.

HIDS/Syslog

In order to keep track of attempts to access resources on our network, we have implemented a syslog server to pull logs from various systems to one central location. We will utilize this system to gather data for intrusion reports.

The server is also running an OSSEC manager plus the ELK stack (Elasticsearch, Logstash, Kibana) in order to monitor/visualize HIDS alerts from all systems in real time. This is an effective measure to stay up to date and aware of events occurring on the systems.

Certificate Authority

In order to better secure our web systems, we have set up our own certificate authority. The Authority is on a standard Debian installation and is not publicly available.

SCADA Infrastructure

Defending SCADA systems has several challenges that are stand out in contrast to classical information security. Modern day IT security has matured to include open standards and strong method of encrypting and authenticating data, users, etc. SCADA systems are often proprietary and the protocols used do not have encryption or authentication. Security must be wrapped around these systems and protocols in order to mitigate their inherent risks.

Our SCADA systems include a proprietary Windows executable HMI application and a proprietary PLC that runs on a Raspberry Pi. These devices communicate using the Modbus TCP protocol. These systems were very insecure out of the box. The HMI application lacks any form of authentication. In addition, the Modbus TCP protocol does not encrypt data and provides no authentication mechanism. This means that if an attacker could access the web server port on the HMI system or connect to the Modbus TCP port on the PLC, then they would have the ability to shut down the power system.

We solved this problem by creating an HMI Proxy system, an EXE server which actually runs the HMI application, and by moving the PLC onto the SCADA network.

HMI Proxy

The HMI Proxy system runs an Nginx web server that act as a reverse proxy. When a user from the LAN accesses this system, they are presented with a HTTP basic authentication prompt in their web browser. This connection will be secured with TLS/SSL by using a certificate from our CA server. After successfully entering their Employee credentials, they will then be served the web page of the HMI application which is running on the EXE server. This allowed us to add authentication to the HMI application without modifying the binary executable.

EXE Server

The EXE Server runs the proprietary HMI application unmodified. The other system that is allow to connect to the web server created by the HMI application is the HMI Proxy.

Raspberry Pi PLC

The Raspberry Pi PLC originally was on the competition network which is the equivalent to having it on the internet. This is a very bad place for a PLC, which uses protocols with no security mechanisms, to be. We have brought this device inside of our SCADA network to prevent incoming connections from system on other networks. Further we have implemented a iptable firewall rule on the device itself to prevent any systems except for the EXE server from connecting to the Modbus TCP port(502).