ISU White Team Documentation for ANL 2017 CDC

For additional support or questions about this document email: anlcdc2017@iastate.edu.
Note: Emails should be sent from outside of the competition network.

Document Revision 1

Defensive Strategies

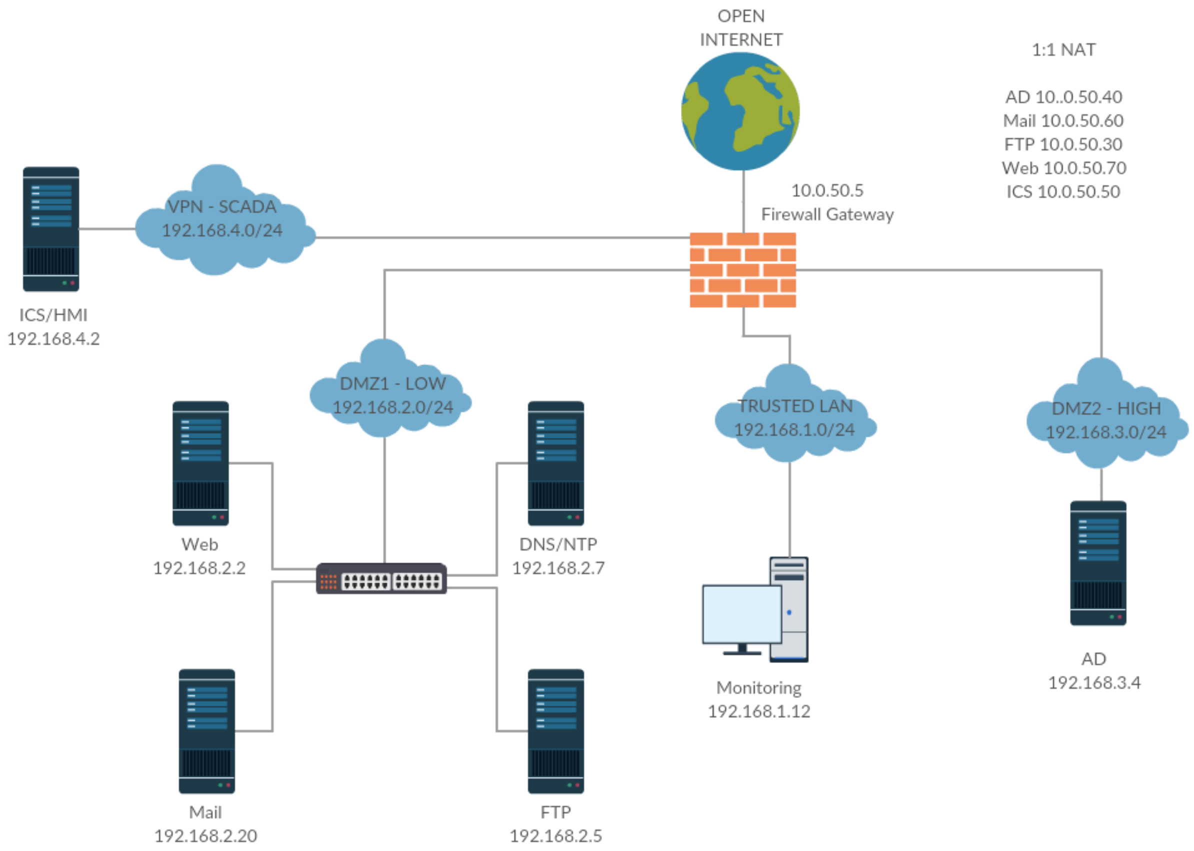
The water and electric industrial control systems (ICS) systems were considered to have the highest impact to our network. However in order to properly control access to these systems our active directory (AD) was also considered a high impact system. We've segmented our network into four corresponding risk sectors: SCADA, HIGH, LOW, and TRUSTED. The ICS was placed alone in the SCADA segment, AD was placed in the HIGH sector, Mail, Web, DNS, NTP, and File server services were placed in the LOW sector, with the remaining monitoring services placed in the TRUSTED sector. A firewall acts as a gateway to the network and routes traffic to the appropriate sector. Monitoring services are run on each network segment and directed to the TRUSTED sector for analysis and real time situational awareness. All servers use host-based firewalls to ensure that they can only talk to each other where necessary.

Aside from segmentation of critical services, we have taken a defense in depth approach to security. Every machine's users, services, and current software configurations have been scrutinized and reduced to the minimal necessary requirements. In addition when tighter access controls and sandboxing can be applied is has been.

To verify our network configuration we have mapped and footprinted all exposed services.

Finally, we have educated and cross trained team members so that every service has two capable network operators. Additionally, each team member has been informed of the social risks of discussing our network and giving access to any of our hardware or documents.

Network Diagram



Active Directory

Operating System

Windows Server 2008 R2

Key Software

Active Directory (domain controller), LDAP

Network Configuration

Local IP Address: 192.168.3.4
External IP Address: 10.0.50.40
Netmask: 255.255.255.0
Exposed Ports: 389 (ldap)

Risk Assessment / Security Implications

The Active Directory domain controller is responsible authenticating users and managing user roles and permissions through our network. It is often one of the most sought after targets for attackers on any network because it can be used to compromise any service depending on it for authentication. In our network this includes the SCADA ICS/HMI and web server services.

Special Security Measures

Since the AD service is so critical to the security of our network, we placed it in its own network segment away from all other services.

Web Server

Operating System

Debian Linux 3.16

Key Software

Docker, Drupal, SQLite, Apache2

Network Configuration

Local IP Address: 192.168.2.2
External IP Address: 10.0.50.70
Netmask: 255.255.255.0
Exposed Ports: 80 (http), 22 (ssh)

Risk Assessment / Security Implications

Traditionally web servers offer a large attack surface area that tend to become entry points into the network. The web server was running an outdated version of of the popular content management system Drupal, which may contain exploitable vulnerabilities. The Drupal site authenticates users through LDAP and so it must have access to services provided by the Active Directory. The site also provides a web browsing interface into the SAMBA file system for hosting employee documentation, which means the web server must be able to access services by the File Server. Finally, the web server provides some access to the HMI console that controls the ICS.

If an attacker is able to compromise the web server it may serve as a foothold for attacks against AD, Mail, DNS/NTP, File Server, and even potentially the ICS/HMI services.

Special Security Measures

SSH was hardened to prevent blank passwords or root logins and fail2ban was installed to prevent brute-forcing of login credentials. Each user is jailed/restricted to their login folder and other ACLs were configured to ensure the principle of least-privilege.

MySQL to SQLite Migration

Originally the Drupal install stored content in a MySQL database. We felt we could reduce our attack surface area by migrating the MySQL drupal database to an SQLite database (which simply consists of a local file). The migration task was not simple, but was achieved by first dumping the drupal database in the MySQL server, dropping table entries for cache, watchdog, and sessions, and finally migrating the database to SQLite with the mysql2sqlite conversion script. Additional Drupal modules module_missing_message_fixer and variablecheck were used to fix the few outstanding errors.

Drupal Patching

Since the version of Drupal running the site was outdated and showed signs of compromise (multiple C99 shells were found during an audit) we decided to patch the Drupal installation to the latest version of Drupal. To do this we version controlled the entire /var/www/html web directory in a private Github repository. In a local VM we installed a fresh version of the latest Drupal 7.x release and incrementally search for and installed the latest versions of the compatible 7.x Drupal modules that were installed on the original image. Finally the custom Nexus child theme was transferred over to the new installation. Using the version control system we were able to determine exactly what changed between the releases and difference out the malicious modifications resulting from the previous compromise. The version control system was then used to create a patch to clean the original Drupal installation.

Drupal Hardening

We further hardened the Drupal installation by leveraging several popular security modules designed to audit and provide additional security measures to Drupal installations. The table below provides a brief overview of the added functionalities.

Module	Functionality
captcha	Injects form catpchas to prevent spam and automated attacks against forms.
flood_control	Limits failed logins by IP, username, and also limits sending emails. Flood control can be used to prevent small DOS attacks.
seckit	Adds security protections for XSS, CSRF, clickjacking, and various strict http header settings.
autologout	Adds automatic session timeouts for users based on roles.

session_limit	Adds support for defining a maximum number of login sessions. If a maximum is reached the oldest session is invalidated and the user is notified that their account may be compromised.
acl	Adds support for restricting access to content based on user roles.
paranoia	Adds Drupal specific protections (ex: prevents editing the administrator user), prevents execution of PHP pages aside from Drupal pages.
security_review	Provides several security audit checks and recommendations for how to fix issues (ex: permissions on writable upload directories).

Docker Sandboxing

To further isolate the Drupal installation from the host OS we decided to use Docker to sandbox the web services from the host OS. This way if the Drupal web server is compromised the attacker will only be gaining access to the Docker sandbox and will have to figure out how to escape the sandbox before he can compromise the host.

As described by Wikipedia:
“Docker provides an additional layer of abstraction and automation of operating-system-level virtualization on Windows and Linux. Docker uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent ‘containers’ to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines.”

To host the web application inside of Docker we created a linux container running PHP with the necessary modules for SQLite and LDAP. We then copied the hardened Drupal installing running with SQLite into the Docker container. The Docker container is launched with a port mapping from the containers hosted ports to the ports exposed by the host machine. We also define two shared data volumes for /var/www/html/IT and /var/www/html/SAMBA_share which map to the host machine’s file server mounts. These mounts are mounted as read only shares, since the web server only needs read access to the shares. Since an attacker with access to the host machine could gain access to the SAMBA password used to mount the share (they are stored in plaintext), the sandboxing through Docker would allow the webserver to be compromised without directly compromising the SAMBA share passwords.

Finally, an added benefit of Docker is that a container can be used to spawn multiple instances of the service. We began developing an in house web application firewall called FWAF (formal web application firewall) that provides formal methods based model checking to check assertions made about an application and enforce stateful transitions in the given finite state machine model. We planned to use this application firewall to provide an additional layer of security and redirect detected malicious activity to a honey pot version of the site contained in the second Docker instance, but as of the time of this writing this system has not yet been fully implemented.

HMI/ICS

Operating System

Raspbian GNU/Linux 8

Key Software

Python Flask (HMI web server), SSHd (ssh server)

Network Configuration

Local IP Address: 192.168.4.2
External IP Address: 10.0.50.50
Netmask: 255.255.255.0
Exposed Ports: 80 (http), 22 (ssh)

Risk Assessment / Security Implications

The ICS/HMI machine is perhaps the most valuable target for attackers. It controls the power and water industrial control systems.

Special Security Measures

Since the HMI interface does not provide any authentication mechanisms we rewrote the python Flask web application to authenticate users via LDAP queries. If users successfully authenticate they are assigned a role based session that allows them to either check ICS statuses or manipulate control systems depending on their level of access. Every URL endpoint aside from a newly introduced login page in the application is restricted to unauthenticated sessions.

SSH was hardened to prevent blank passwords or root logins and fail2ban was installed to prevent brute-forcing of login credentials. Each user is jailed/restricted to their login folder and other ACLs were configured to ensure the principle of least-privilege.

The HMI device was connected into the internal network using OpenVPN and segmented on to it’s own network called SCADA.

Mail

Operating System

Ubuntu 16.04 Server

Key Software

Dovecot (IMAP/POP3), Postfix (mail server), SquirrelMail (web interface)

Network Configuration

Local IP Address: 192.168.2.20
External IP Address: 10.0.50.60
Netmask: 255.255.255.0
Exposed Ports: 80 (http)

Risk Assessment / Security Implications

If the mail server is compromised an attacker could potentially pivot to attack the web server, DNS, NTP, and file server services. Additionally sensitive user email data may become compromised if an attacker gains access to the mail server.

Special Security Measures

Setting up mail servers correctly is a tricky task. To combat this risk we have place our mail server in a low risk network segment to isolate it from other high risk services.

We provide a SquirrelMail web client for users to access email so we only need to expose port 80 to allow user access. Since all services leveraging mail are within the network and the mail services are not required ports in the rules documentation we do not expose the mail protocols outside the network (sending and receiving mail is an internal service only; users cannot send or receive mail to or from email accounts outside of the network, however users can remotely access mail to send and receive emails to and from other employees within the network).

File Server

Operating System

Ubuntu 14.04LTS Server

Key Software

vsftpd (FTP/FTPS), Samba (file sharing), SSHd (ssh server)

Network Configuration

Local IP Address: 192.168.2.5
External IP Address: 10.0.50.30
Netmask: 255.255.255.0
Exposed Ports: 21 (FTP/FTPS), 22 (SSH)

Risk Assessment / Security Implications

The file server is on the same network segment as mail, web, DNS, and NTP, which could become potential targets if an attacker is able to pivot from the file server and the firewall stops enforcing host based access controls within the network. Aside from gaining a foothold into the network the file server contains sensitive documentation for employees discussing the network configuration, user passwords, and defensive strategies.

Special Security Measures

Since only the web server needs to read from SAMBA, we configured SAMBA to be read only so that it can only read and list files. In order to allow users to upload files we allow FTP to list files and write files (but not read files). To combat the leakage of sensitive data from the files contained on the file server, we encrypt the shared documents and provide users with instructions for decrypting downloaded and encrypting uploaded documents. Furthermore, the FTP server supports FTPS and will upgrade the connection to a TLS-secured session if the client's software can support it.

The FTP server is VsFTPd, and it is configured to chroot itself into a secure directory and drop all root privileges. Furthermore, the service is "jailed" through the use of AppArmor policies to ensure that it is not able to read any files that it should not have access to. Each user can login and is jailed to the appropriate directory.

Samba is configured to require SMB signing and unnecessary shares/services were disabled such as printer support. The files Samba and the FTP server share are located on a separate partition that is mounted with the nosuid and noexec flags set to enhance security.

SSH was hardened to prevent blank passwords or root logins and fail2ban was installed to prevent brute-forcing of login credentials. Each user is jailed/restricted to their login folder and other ACLs were configured to ensure the principle of least-privilege.

Users are provided access to encryption passwords by calling a toll-free phone number (1-877-844-1072) and entering their assigned badge number concatenated with a secret user assigned PIN number. As an added layer of security the in-house developed phone system (using the Twilio APIs) only issues passwords to decrypt files that are needed for the user's role and the documents needed at the given time of day.

DNS/NTP

Operating System

OpenBSD 6.0

Key Software

ISC BIND (DNS Server)
NTPD (NTP Daemon)

Network Configuration

Local IP Address: 192.168.2.7
Netmask: 255.255.255.0
Exposed Ports: 53 (DNS), 123 (NTP), 22 (SSH)

Risk Assessment / Security Implications

If the DNS server is compromised, DNS entries can be changed and potentially break several servers that depend on this. It can also be used to man-in-the-middle servers which are tricked into thinking a server is at a different IP address.

Pivoting to other servers is limited as this server runs a host-based firewall with the OS running in securelevel=2 meaning the firewall rules cannot be altered so even if a root-compromise occurs, it would be extremely difficult to alter the rules.

Special Security Measures

All network services are run using unprivileged accounts to minimize impact if the program is exploited.

Firewall rules limit the access of the device to an “as needed” basis. The firewall also limits the amount of packets that come in and offenders are put on a ban list.

SSH was hardened to prevent blank passwords or root logins, as well as use public-keys only for logging in.

Firewall

Operating System

PfSense 2.3.3 x64

Key Software

OpenVPN
Snort IDS

Network Configuration

External IP: 10.0.50.5
Internal IP: 192.168.{1,2,3,4}.1/24
Exposed Ports: 22 (SSH), 1194 (OpenVPN)

Risk Assessment / Security Implications

If the firewall is compromised, the attacker gains full access to our LAN, as well as the ability to sniff and tamper with all of the traffic on the LAN as it is in the middle of everything. It would have full network control and could pivot to attack devices as well as denying network connections to all of our servers.

Special Security Measures

The firewall has been configured to only allow SSH key logins and no passwords. Finely-tuned rules have been implemented to ensure devices are only able to talk to each other when necessary.

Monitoring

Operating System

Ubuntu 16.04 x64

Key Software

Greylog (Syslog pool)

Network Configuration

IPv4: 192.168.1.12/24
Exposed Ports: 9000 (HTTP Web UI)

Risk Assessment / Security Implications

An attacker could read all of the logs from our servers and snoop on any sensitive data contained within. They could also send spoofed log files in order to trick us into thinking something was happening. Pivoting to other devices will not be possible as the firewall rules will not allow this and it is on its own separate LAN.

Special Security Measures

None.