SECURITY AND RISKS

Rob Kraft and Brian Treff

SHARPENyourTOOLS

lucity™ act16

AGENDA

- What are we trying to protect?
- What are the Threats?
- Risks for organizations exposing nothing to the Internet
- Risks exposed by Open Ports
- Risks exposed by Web Apps
- Risks exposed by REST APIs/GIS
- Risks exposed by Tablet and Phone Apps
- Defense In Depth

ASSETS

- Assets are the things we are trying to protect
  - Protect data from alteration
  - Protect data from destruction
  - Protect data from discovery
  - Protect credentials of people and processes to data and computers
  - Protect computers from infection which can lead to
    - Any of the above
    - Computer used to store files for hackers
    - Computer used as compute resource for hackers
    - Computer used to launch attacks elsewhere around the world
  - Protect your reputation

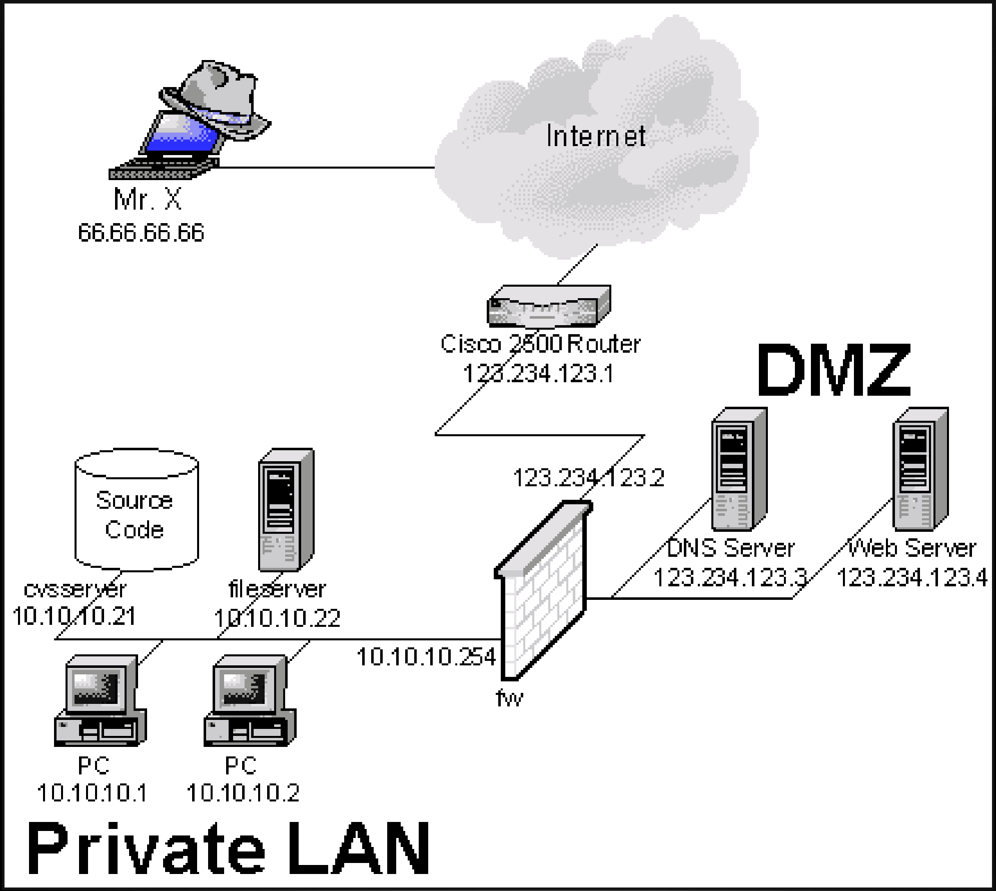
THREATS

- Enemy states
- Hacktivists
- Someone looking to gain a reputation as a hacker
- Disgruntled citizen
- Disgruntled employee

RISKS WITH NO OPEN PORTS

- Visiting infected Web Sites
- Downloading/installing infected software
- Phishing (usually from email) and Spear Phishing
  - Ransomware
  - Malware
  - Keystroke Loggers
  - Zombies
  - Rootkits
  - Zero Day Attacks
- Faulty or misconfigured router
- WIFI

RISKS WHEN YOU OPEN PORTS



RISKS WITH OPEN PORTS

- Port Scanning from anywhere in the world. Hard to detect.
  - Then they can often determine what OS, App, and Version is listening on the port.
- Possible Vulnerabilities
  - Unpatched software
  - Poor credentials
  - Improper/Insecure Configurations
  - Buffer overflows
  - Denial Of Service (DOS) attacks (DDOS attacks)
- Easy for Hackers to Exploit
  - Automated Tools updated with latest flaws

RISKS WITH WEB APPS

- Ports exposed on the internet
- Web server attack/crash/flood
- SQL Injection
  - Depends on perms allocated to account logging into SQL Server (Connection string)
  - Depends on perms allocated to service running SQL Server/Oracle
- Javascript injection (XSS)
  - Depends on UI Rendering that does not sanitize
- Cross Site use of Lucity App (CSRF)
- Clickjacking
- Impersonation, credentials theft

RISKS WITH REST APIS/GIS

- SQL Injection/Javascript injection
- Web server attack/crash/flood
- Cross Site use of Lucity App
- Web Citizen REST API (Public) concerns
  - Very locked down. Few endpoints.
- Mobile REST API concerns
  - Locked down some. No “Filters”.
- Internal REST API concerns
  - Not locked down. Filters allowed.

RISKS WITH TABLET AND PHONE APPS

- Theft/Loss of the Device
  - Thief using the apps
  - Thief examining the disconnected data
- Wireless network risks
  - Sniffing of data between the tablet and REST API
  - "Fake" public sites
  - Devices autoconnect to "home"
  - SSL/TLS – Man In The Middle (MITM)
    - MITM Analogy – a restaurant waiter
  - Cellular is more secure than wifi
- New Warehouse App in Lucity 2015R2

MORE RISKS

- Things you send offsite like backups. Are they encrypted?
- VPN connections into your organization. Are they reviewed?
- Data you host in the cloud?
  - Dropbox, Google Docs, OneDrive, ShareFile
- Data coming in to your organization:
  - Collected from Scada systems
  - Collected from other automated systems
- Physical access to your servers and networks

DEFENSE IN DEPTH

- Education (especially about phishing)
- Passwords
- Limited Permissions
- Change Default Configurations
- Keep Software Up To Date
  - OS, Browsers, Apps, Firmware
- Firewalls/Routers
- De-Militarized Zones (DMZ)
- Virtual Private Networks (VPN)s
- Intrusion Detection/Prevention Systems (IDS/IPS)
- Anti-virus detection
- Periodically Review Configurations and security
- Have an Incident Response Plan
- Wireless Routers
  - turn them off
  - Use WPA2, not WPA and definitely not WEP

LUCITY SOFTWARE

- SQL Injection
  - Stored Procedures to Parameterize
  - All Dynamic SQL is funneled through the same function that tests for:
    - unmatched quotes,
    - unexpected keywords (SQLBlackList),
    - nonAscii chars
  - The database account does not have permission to alter table structures
  - Mobile/REST blocks 'filter' parameter
- Javascript injection
  - Sanitize data before displaying in HTML
- Clickjacking
- Database Backups
  - Passwords are hashed with Bcrypt and unique salts
  - Encrypt entire backup (Enterprise Edition of SQL Server Pre SQL 2014)
- Password Management for Tablets, SaaS, and SSL Internet Implementations
  - Min Password Length, Password Complexity, Password Expiration, Password History
    - Future – User Must Change Password, more

TERMS

- Threats
- Attack Vectors
    - Vulnerabilities
        - All
            - Known
            - Unknown
- Assets – What we are trying to protect
- Risks – Probability * Criticality
- Data in Transit/Data at Rest