

# Securing Legacy Apps – Worth the Effort?

Michael Solomon

29 June 2016



**Yes. Well, it depends.**

It depends on thoughtful assessment of your application and its vulnerabilities

# Agenda

- What is a legacy application?
- How are they vulnerable?
- What can we do about it?
- Where can I get help?



Legacy application

Software **designed** without  
considering **security**.

You do have AT LEAST one legacy application, right?



Legacy security often relied on physical barriers.  
Physical controls protected data.

# Common legacy security vulnerabilities

## Threats developed after the application (ongoing)

- Lack of patching for available fixes
- OS or infrastructure may be past EOL

## Trend to expose applications designed for internal use

- Data crossing trust boundaries

## Aging environments

- OS
- Libraries
- Infrastructure
- Are latest patches compatible with application?

## Relaxed identification and authentication

- Best practices have changed
- Reduced need to validate client identity

## Decentralized

- Authentication / Access control / Auditing

These are just the most common reasons that legacy applications have security vulnerabilities.

# Legacy client types

## Terminal systems

- Mainframe
- TTY terminals
- OpenEdge Character (CHUI) client

## Client/Server

- Workstation
- Networked connection to database
- OpenEdge GUI client

## Browser

- Internet application
- Runs in web browser
- OpenEdge WebSpeed

# Terminal system application vulnerabilities

Authentication often depends on OS

Authorization based on file system permissions

\*\*\* Limited or nonexistent encryption capability \*\*\*  
in-transit/at rest

No local input data validation

Often dependent on physical terminal characteristics (perhaps for input validation)

Due to historical physical separation security

- other areas may be sloppy (i.e. hardcoded userids/passwords)



# Client/Server application vulnerabilities

Some applications offload too much data to clients for efficiency

Server DB is exposed to non-authorized connections

Deployment can be problematic and slow down distribution of patches

Potential vulnerabilities for older Client/Server software

Session hijacking

Harder to administer well

- Many insecure defaults are in the wild

## Browser application vulnerabilities

- Session hijacking
- Exposure of server components (DB/AppServer)
- Higher reliance on network/server security
- More difficult to handle per-call authentication/authorization

# What does it take to secure legacy applications?

## Inventory all legacy applications

- Description
- versions in use
- Dev history
- Footprint
- Requirements
- Responsible person

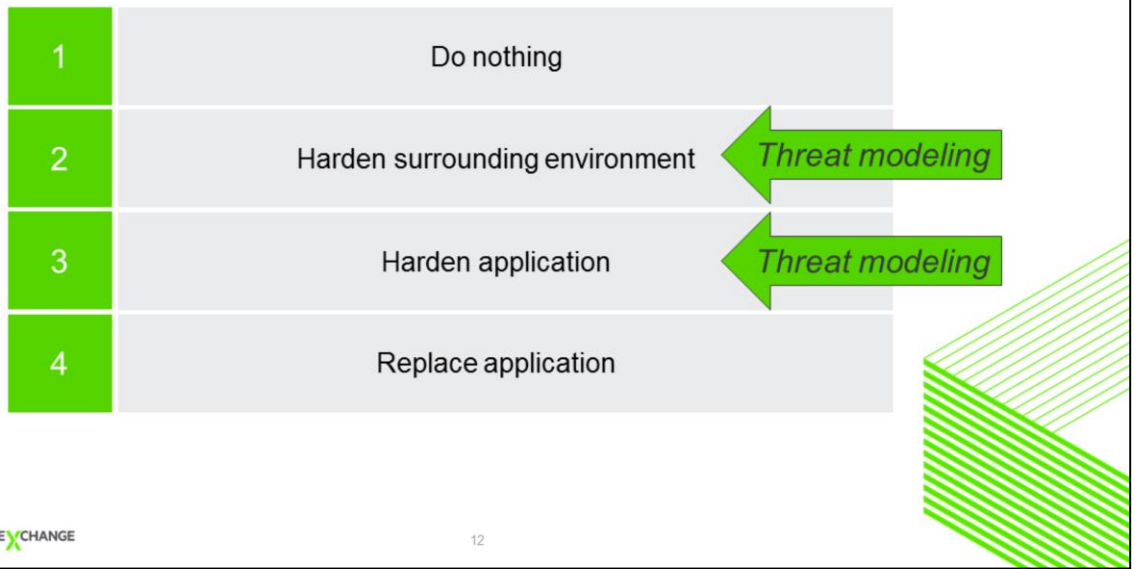
## Risk assessment (limit scope for expediency if necessary)

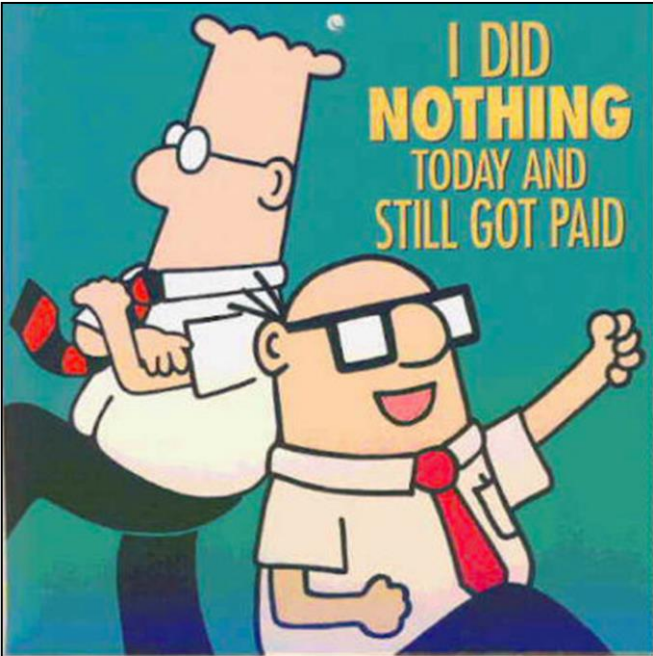
- Sensitive data access
- Business function criticality
- Compliance
- Attack surface

## In-depth analysis of "at high risk" systems

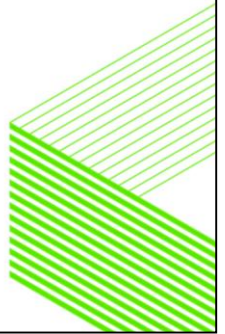
## Mitigate high priority risks

## Mitigation options





- May be the best choice
- Risk assessment provides response recommendations
- Better than wasting resources on ineffective controls
- Choose this option
  - Don't default to it



Manage risk with threat modeling





## How to threat model

What are you building?

What can go wrong?

What are you going to do about it?

Check your work



# What are you building?

## Create a system model

- Abstracts away the details

## Diagrams are key

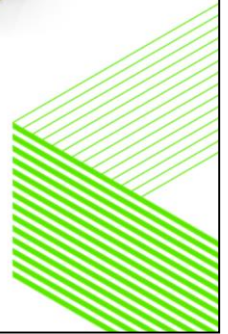
- Mathematical models are rare in industry

## Primary focus in threat modeling

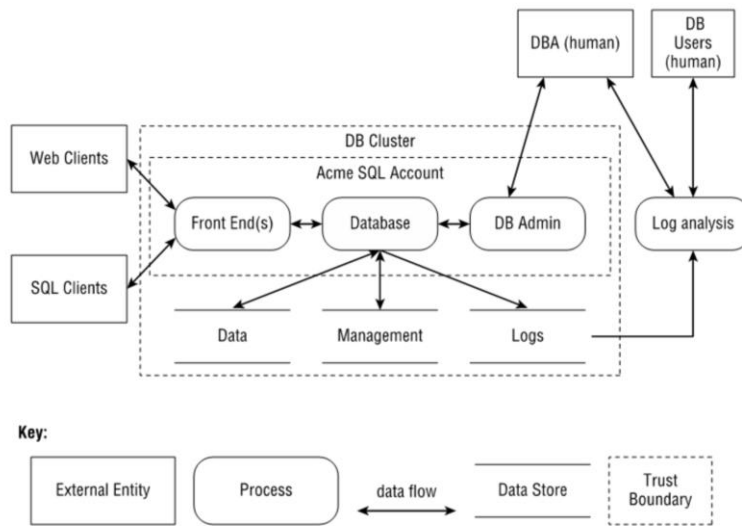
- Data flows
- Threat boundaries

## Common diagram types

- Data Flow Diagrams (DFD)
- Swim Lanes
- State machines



# Data Flow Diagram



Developed in the early 70s, and still useful

Simple: easy to learn, sketch

Threats often follow data

Abstracts programs into:

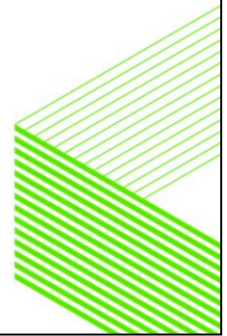
Processes: your code

Data stores: files, databases, shared memory

Data flows: connect processes to other elements

External entities: everything but your code & data. Includes people & cloud software

Trust boundaries now made explicit



## What can go **wrong**?

Fun to brainstorm

Mnemonics, trees or libraries of threats can all help structure thinking

Structure helps get you towards completeness and predictability

STRIDE is a mnemonic

Spoofting, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege

Easy, right?

# STRIDE

S – Spoofing

T- Tampering

R – Repudiation

I – Information disclosure

D – Denial of Service

E – Elevation of privilege

Threat	Property Violated	Definition	Example
<b>Spoofing</b>	Authentication	Impersonating something or someone else.	Pretending to be any of Bill Gates, Paypal.com or ntdll.dll
<b>Tampering</b>	Integrity	Modifying data or code	Modifying a DLL on disk or DVD, or a packet as it traverses the network
<b>Repudiation</b>	Non-repudiation	Claiming to have not performed an action.	"I didn't send that email," "I didn't modify that file," "I <i>certainly</i> didn't visit that web site, dear!"
<b>Information Disclosure</b>	Confidentiality	Exposing information to someone not authorized to see it	Allowing someone to read the Windows source code; publishing a list of customers to a web site.
<b>Denial of Service</b>	Availability	Deny or degrade service to users	Crashing Windows or a web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole.
<b>Elevation of Privilege</b>	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but going from a limited user to admin is also EoP.

# Using STRIDE

How can each STRIDE threat can impact each part of your model

- How could an attacker tamper with this part of the system?

Make it easier

- Elevation of Privilege Game
  - <https://www.microsoft.com/en-us/sdl/adopt/eop.aspx>
  - <https://www.thegamecrafter.com/games/elevation-of-privilege>
- Attack Trees
- Experience

Track issues as you find them

- Track assumptions too



The background of the slide is a dark teal color with several lighter teal diagonal stripes that create a sense of movement or a path leading towards the right. The stripes are parallel and spaced evenly.

What are you going to **do** about it?

## Threats and assumptions

### For each threat

- Fix – remove functionality
- Mitigate
- Accept – Be careful about accepting customer risk
- Transfer – License agreements, TOS

### For each assumption

- Check
- Reconsider wrong assumptions





# Ways to mitigate threats

Threat	Mitigation Technology	Developer Example	Sysadmin Example
Spoofing	Authentication	Digital signatures, Active directory, LDAP	Passwords, crypto tunnels
Tampering	Integrity, permissions	Digital signatures	ACLs/permissions, crypto tunnels
Repudiation	Fraud prevention, logging, signatures	Customer history risk management	Logging
Information disclosure	Permissions, encryption	Permissions (local), PGP, SSL	Crypto tunnels
Denial of service	Availability	Elastic cloud design	Load balancers, more capacity
Elevation of privilege	Authorization, isolation	Roles, privileges, input validation for purpose, (fuzzing*)	Sandboxes, firewalls





## Check your work

Quality assurance

Check that you covered all the threats & assumptions

Check that each is covered well



**And if threat modeling doesn't  
meet your needs ...**

Sometimes you just can't mitigate enough vulnerabilities in legacy applications.



Replace software when mitigation cost is too high and replacement software meets needs



The new Solomon Consulting portal coming soon.

Ready

Set

Secure

[www.solomonconsulting.com](http://www.solomonconsulting.com)

## References

- <https://buildsecurityin.us-cert.gov/articles/best-practices/legacy-systems/assessing-security-risk-in-legacy-systems>
- <http://www.computerworld.com/article/2474158/application-security/4-risks-from-legacy-applications.html>
- <http://www.csoonline.com/article/2139382/data-protection/forgotten-risks-hide-in-legacy-systems.html>
- <https://securityintelligence.com/what-are-the-risks-of-legacy-infrastructure/>
- [http://www.codingthearchitecture.com/2015/03/07/security\\_concerns\\_for\\_legacy\\_systems.html](http://www.codingthearchitecture.com/2015/03/07/security_concerns_for_legacy_systems.html)
- <http://www.datacenterjournal.com/securing-legacy-applications-modern-threats/>
- [https://www.owasp.org/index.php/OWASP\\_Supporting\\_Legacy\\_Web\\_Applications\\_in\\_the\\_Current\\_Environment\\_Project](https://www.owasp.org/index.php/OWASP_Supporting_Legacy_Web_Applications_in_the_Current_Environment_Project)

