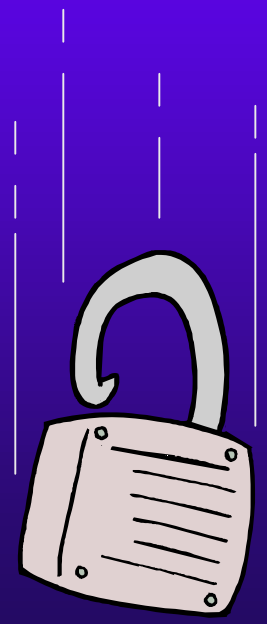
Oded Sacher
Software Design Engineer
Microsoft Haifa R&D Center

Building Secure Software

Agenda

- The need for secure systems
- The proactive security development process
- Security principals to live by
- Threat modeling
- Conducting a security push



Definitions

- A secure product – A product that protects the confidentiality, integrity, and availability of the customer's information, and the integrity and availability of processing resources, under control of the system owner.
- A security vulnerability – A flaw in a product that makes it infeasible to prevent an attacker from usurping privileges on the user's system, regulating its operations, compromising data, or assuming ungranted trust.

The Need For Secure Systems

- Intercommunication in the Internet era (servers, desktops, cell phones, pocket devices) introduces business opportunities and exposure to harsh environment.
- Wild Wild Web – W2K honey pot.
- Trustworthy computing – Bring software to the same trust level as phone, electricity…

Proactive Security Development Process

- Security education
- Design phase – Determine security goals, Security is a product feature, Threat modeling
- Development phase – Coding guidelines, Code review, learn from old defects.
- Test phase – Test that the system design and code can withstand attack. Think evil.
- Shipping/Maintaining phase

Security Principals To Live By

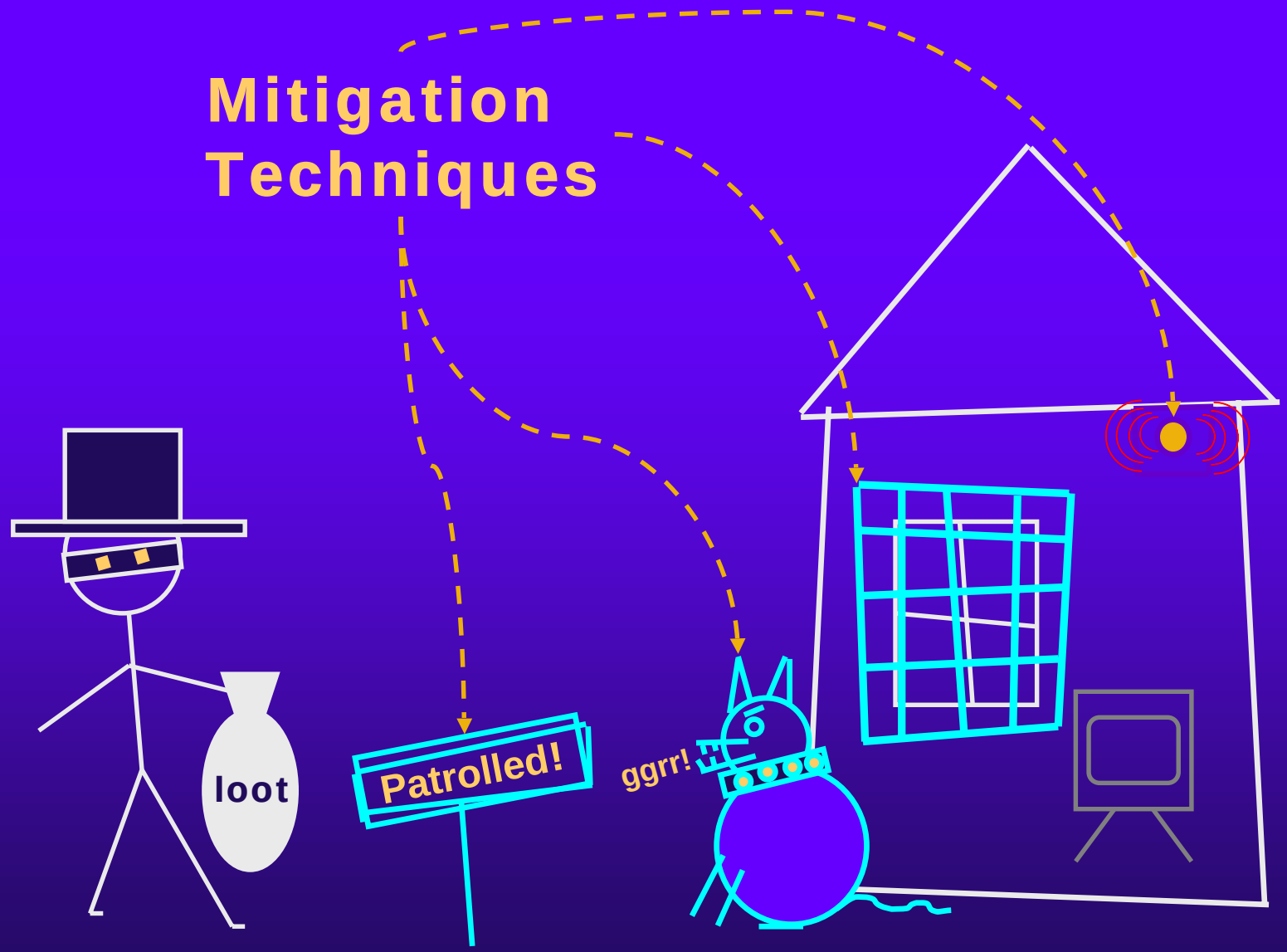
- Learn from mistakes
- Minimize the attack surface
- Employ secure defaults
- Use defense in depth
- Use least privileges
- Backwards compatibility and security
- Assume external systems are insecure
- Plan on failure, and fail to a secure mode
- Don’t depend on security by obscurity
- Be ware of mixing code and data

Threat Modeling

- You cannot build a secure product unless you understand the threats
- Security based analysis of the system
- Determine and prioritize the security risks to the system
- Propose mitigation techniques


Threat
loot
Asset
Vulnerability

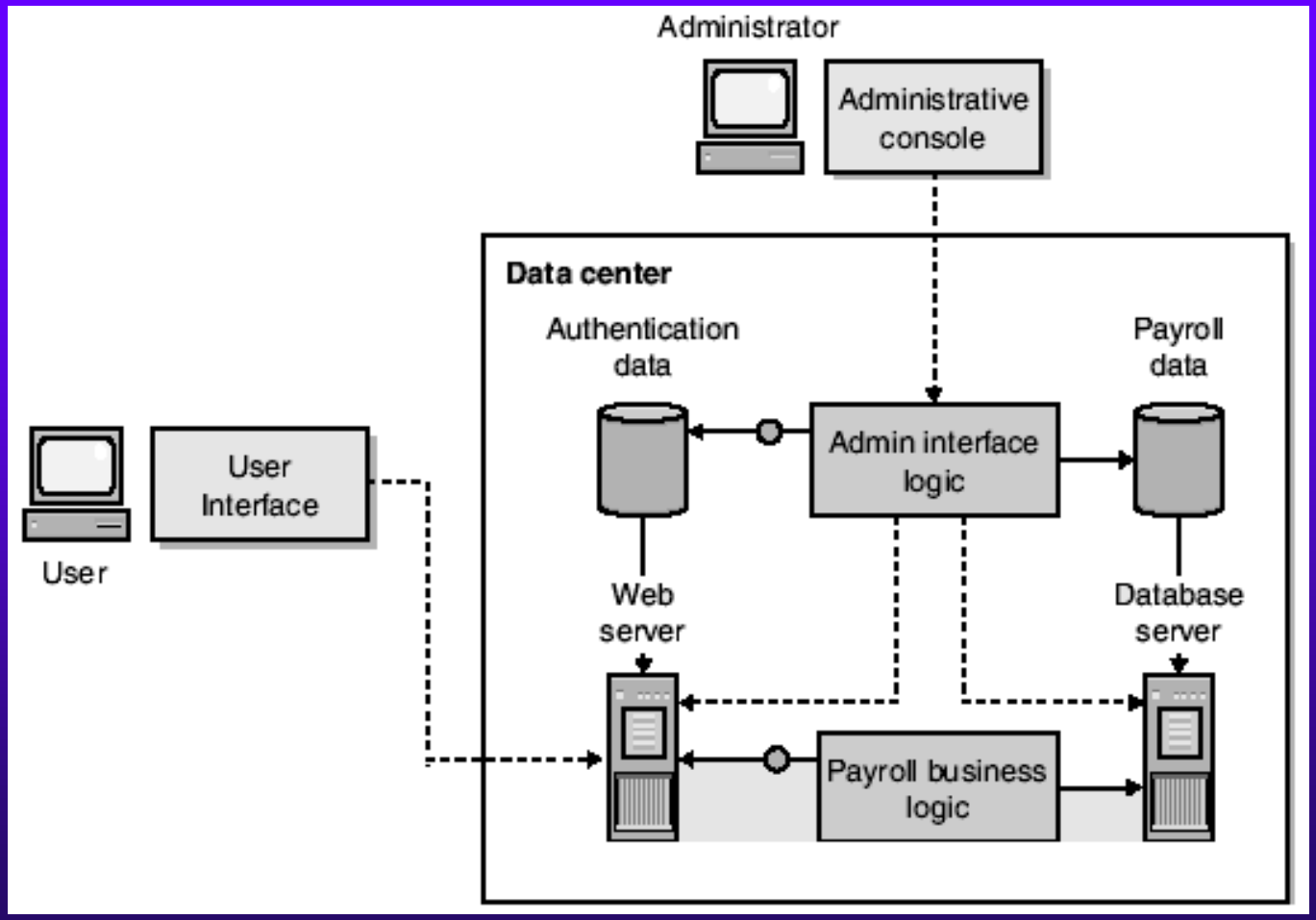
Mitigation Techniques



The Benefits Of Threat Modeling

- Helps you find security bugs
- Helps new team members to understand the product security design
- Helps other product team that build on your product
- Helps testers to build an effective security test plan

Payroll Application Sample



Determine The Threats

- Look at the components and ask questions like:
  - Can a nonauthorized user view confidential network data?
  - Can an untrusted user modify the data in the database?
  - Can someone deny valid users service from the application?
  - Can someone take advantage of a feature or component to raise their privileges to that of an Administrator?

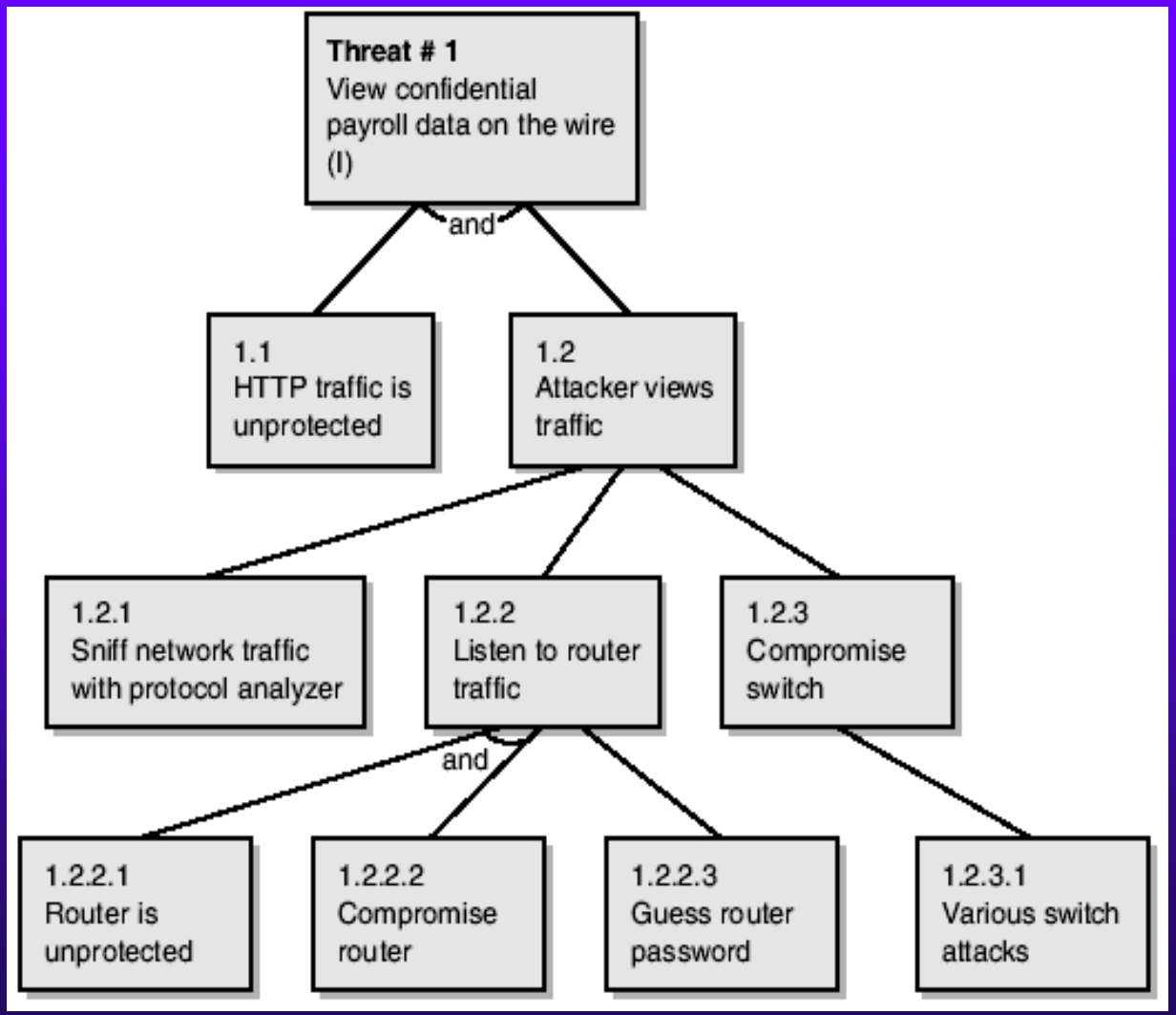
Use STRIDE To Categorize Threats

- **S**poofing identity – Allow an attacker to pose as another user, or allow a rough server to pose as a valid server.
- **T**ampering with data – Malicious modification of data.
- **R**epudiation – Deny performing an action without other parties having any way to prove otherwise.
- **I**nformation disclosure – Exposure of information to users who are not supposed to have access to it.
- **D**enial of service – Deny service for valid users.
- **E**levation of privileges – Unprivileged user gains privileged access and thereby can compromise the entire system.

Evaluate Threats Risk

- $Risk_{CO}$ = Criticality * Likelihood of occurance
- Use DREAD
  - **D**amage potential
  - **R**eproducibility
  - **E**xploitability
  - **A**ffected users
  - **D**iscoverability
- $Risk_{DREAD}$ = AVERAGE (D, R, E, A, D)

Payroll Application Threats



D	8
R	10
E	7
A	10
D	10
Overall	9

Mitigation Techniques

Threat type	Mitigation technique
Spoofing identity	Strong authentication, Protect secrets, Don’t store secrets
Tampering with data	Authorization, Digital signatures
Repudiation	Digital signatures, Time stamps, Auditing
Information disclosure	Authorization, Encryption, Protect secrets, Don’t store secrets
Denial of service	Authentication, Authorization, Filtering, Throttling
Elevation of privileges	Run with least privileges

Security Push

- Starting late 2001, MS initiated several security pushes.
- Most significant was (8 weeks) February Windows security push.
- Everyone! stops all non-security activity and concentrate on leveraging the product security.

Security Push Goals

- Raise the security awareness of everyone in the team
- Find and fix issues in the code, and in some instances, the design of the product

Security Push Plan

- Security education
- Threat modeling
- Security code review
- Security testing
- Security documentation
- Triage security bugs

Security Education

- “Writing Secure Code” mandatory reading
- Live and online presentations
- Internal Q&A aliases
- Security push web site
  - All groups plan
  - Threat model samples
  - Security bulletin analysis

The Program Manager's Role

- Get Trained. Read Book.
- Build threat models
- Do whatever it takes to reduce the attack surface
  - Evaluate default installation and configuration
  - Run with lower privileges
- Clearly document security issues in the product, and clean sample code

The Developer's Role

- Get Trained. Read Book.
- Perform security oriented code reviews
  - Secure build options (/GS, /robust)
  - Look for dangerous APIs
  - Verify untrusted input
  - Review all access control lists
  - No secrets in the code
- Use automatic code scanning tools such as PREfix and PREfast

The Tester's Role

- Get Trained. Read Book.
- Think evil. Test that stuff breaks instead that it works.
- Use data mutation tests (RPC attack)
- Run tests as non-admin
- Use Application/Driver verifier
- Review old security bugs resolved as "postponed, by design …"
- Regression tests for security fixes

Summary

- Hackers invest an intensive effort to find and exploit security holes.
- The product team must structure security into all steps of their development process.
- Most important is raising everyone's awareness to SECURITY.
- Questions?