CS615 - Aspects of System Administration

System Security

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://www.cs.stevens.edu/~jschauma/615/

Where/how does 'security' come into play?

Where/how does 'security' come into play?

Lecture 02 (Filesystems, Disks, Storage)

- storage model (DAS, NAS, SAN, Cloud)
- partitions / mount options
- filesystem features (permissions, access control lists)
- DoS on disk space
- firmware compromise on hard drives

Lecture 03 (Software Installation Concepts)

- software package management and updates
- VMs, containers, etc.
- patch management
- package integrity checking

Where/how does 'security' come into play?

Lecture 04 (Multiuser Fundamentals)

- privileges and trust models
- authentication methods, multi-factor authentication
- file access controls
- raising privileges

Lecture 05 / 06 (Networking)

- protocols and visibility of data on different layers
- tcpdump can read all packets
- location of attacker on network implies capabilities
- network censorship

Where/how does 'security' come into play?

Lecture 07 (DNS; HTTP)

- If you control the DNS, you control the domain
- DNS registrars as attack points
- use of DNS as another channel for host verification (SSHFP records)
- trustworthiness of DNS (DNSSEC)

Where/how does 'security' come into play?

Lecture 08 (HTTPS, Monitoring)

- cleartext vs ciphertext
- TLS authentication
- PKI, Certificate Authorities
- protocol downgrade and MitM attacks
- incident detection via events, metrics, and context
- sensitive data in logs
- outsourcing monitoring services

Where/how does 'security' come into play?

Lecture 09 (Writing System Tool)

- automation as a defensive weapon
- using the wrong tool for the job => writing insecure code
- understanding language / framework pitfalls
- simplicity reduces attack surface

Where/how does 'security' come into play?

Lecture 10 (SMTP, Backup and Disaster Recovery)

- email as attack methods (spam, phishing)
- email privacy implications
- SMTP plain text vs. opportunistic encryption
- mail abuse and spam
- recipient and sender authentication, open relays
- disasters include security breaches
- safety of backups (encrypted backups?)

Where/how does 'security' come into play?

Lecture 11 (Configuration Management)

- inherent trust, full control
- CAP theorem may impact security controls

Lecture 12 (Ethics and Social Responsibility)

- privacy and responsibility
- lead by example
- implications of data retention
- transparency
- continuous education

How do we secure a system?

How do we secure a system?

It depends.

(Context required.)

What is security?

```
security

 NOUN:

    Freedom from risk or danger; safety.
```

What is risk?

```
risk

 NOUN:

  The possibility of suffering harm or loss; danger.
```

Suffering harm or loss of *what*?

- access to data

Suffering harm or loss of *what*?

- access to data
- integrity of data

Suffering harm or loss of *what*?

- access to data
- integrity of data
- availability of services

Suffering harm or loss of *what*?

- access to data
- integrity of data
- availability of services
- reputation

Suffering harm or loss of *what*?

- access to data
- integrity of data
- availability of services
- reputation
- monetary loss due to any of the above

Suffering harm or loss of *what*?

- access to data
- integrity of data
- availability of services
- reputation
- monetary loss due to any of the above
- monetary loss due to physical items of actual value

Suffering harm or loss of *what*?

- access to data
- integrity of data
- availability of services
- reputation
- monetary loss due to any of the above
- monetary loss due to physical items of actual value
- ...

How to determine *risk*

"Risk Assessment"

- identify *assets*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*
- identify *vulnerabilities*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*
- identify *vulnerabilities*
- determine *likelihood of damage*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*
- identify *vulnerabilities*
- determine *likelihood of damage*
- estimate *cost of recovery*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*
- identify *vulnerabilities*
- determine *likelihood of damage*
- estimate *cost of recovery*
- estimate *cost of defense*

How to determine *risk*

"Risk Assessment"

- identify *assets*
- identify *threats*
- identify *vulnerabilities*
- determine *likelihood of damage*
- estimate *cost of recovery*
- estimate *cost of defense*

A *risk* is the *likelihood* of a *threat* successfully exploiting a *vulnerability* and the *estimated cost* (or potential damage) both in the short and long term you may incur as a result.

Threat Model

For each system/component/product/service/...

- identify *what* you're protecting
- identify *from whom* you're protecting it
  - identify *goals* of the attacker
  - identify *motivation* of the attacker
  - identify *capabilities* of the attacker
- identify threats you cannot defend against (within this system or in general)

Threat Model

Your adversaries are determined human actors with specific goals.

Imperatives

Constantly seek to reduce your attack surface.
Identify and eliminate attack vectors.

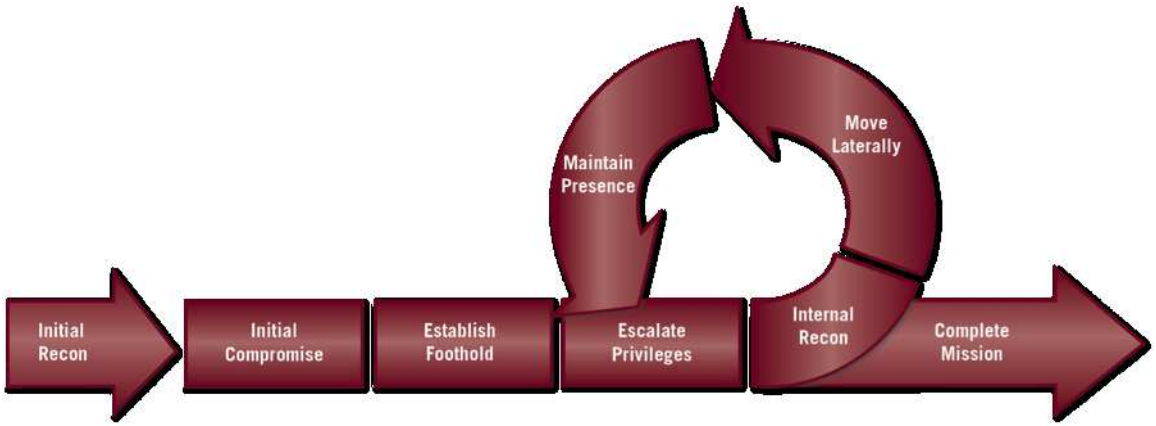
You can't do this alone:
lead by example, seek allies.

Defense in Depth

Security is like an onion:
the more layers you peel away, the more it stinks.

The biggest threat comes from the inside

The biggest threat comes from the inside



http://is.gd/6sREQh

Cryptography

Cryptography can help mitigate *some* of the risks *sometimes*.

Cryptography

Cryptography can help mitigate *some* of the risks *sometimes*.

It may provide security in the areas of:

- Secrecy or Confidentiality
  - *Did/could anybody else see (parts of) the message?*

Cryptography

Cryptography can help mitigate *some* of the risks *sometimes*.

It may provide security in the areas of:

- Secrecy or Confidentiality
  - *Did/could anybody else see (parts of) the message?*
- Accuracy or Integrity
  - *Was the message (could it have been) modified before I received it?*

Cryptography

Cryptography can help mitigate *some* of the risks *sometimes*.

It may provide security in the areas of:

- Secrecy or Confidentiality
  - *Did/could anybody else see (parts of) the message?*
- Accuracy or Integrity
  - *Was the message (could it have been) modified before I received it?*
- Authenticity
  - *Is the party I'm talking to actually who I think it is / they claim they are?*

Cryptography

Note:

- *Authentication* != *Authorization*
- cryptography does not handle authorization
- you generally need all three: confidentiality, integrity, authenticity
- cryptography cannot prevent against incorrect use
  – usability is hard!

Know your threat model!

Basic Security Concepts: Confidentiality

- Alice and Bob agree on a way to transform plain text into ciphertext
- transformed data is sent over insecure channel
- Alice and Bob are able to reverse transformation

Basic Security Concepts: Confidentiality

- Alice and Bob agree on a way to transform plain text into ciphertext
- transformed data is sent over insecure channel
- Alice and Bob are able to reverse transformation

Different approaches:

- secret key cryptography (example: *DES*)
  - Alice and Bob share a secret key
- public key cryptography (example: *RSA*)
  - Alice has a private and a public key
  - data encrypted with her private key can only be decrypted by her public key and vice versa
  - public key can be shared with anybody (via insecure means)

Threats to Confidentiality

- lack of *authenticity*
- key exchange
- key disclosure

Basic Security Concepts: Integrity

In order to protect against forgery or data manipulation, provide some sort of digest or checksum (often a one-way hash). Popular choices:

- `5f4dcc3b5aa765d61d8327deb882cf99`
- `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8`
- `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8`
- `b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86`

Basic Security Concepts: Integrity

In order to protect against forgery or data manipulation, provide some sort of digest or checksum (often a one-way hash). Popular choices:

- `5f4dcc3b5aa765d61d8327deb882cf99` (MD5)
- `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8` (SHA-1)
- `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62 a11ef721d1542d8` (SHA256)
- `b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5 e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a 2ea6d103fd07c95385ffab0cacbc86` (SHA512)

Basic Security Concepts: Integrity

Examples: host based IDS, package manager signatures

Some possible threats:

- collisions in algorithm
- lack of *authenticity* (Where did I get the checksum?)
- lack of *integrity* (Was the checksum tampered to match the (tampered) data?)
- “verification” with compromised tools
- “rainbow tables” / internet search engines allow for easy reverse lookup of un-salted hashes.

Basic Security Concepts: Authenticity

Three general ways of proving that you are who you say you are:

- something you know
- something you have
- something you are

Basic Security Concepts: Authenticity

Three general ways of proving that you are who you say you are:

- something you know
  - secret handshake, password
  - can (easily) be given to and used by somebody else
- something you have
- something you are

Basic Security Concepts: Authenticity

Three general ways of proving that you are who you say you are:

- something you know
  - secret handshake, password
  - can (easily) be given to and used by somebody else
- something you have
  - physical items: smart card, RSA token, ...
  - private keys
  - can (easily) be given to and used by somebody else
- something you are

Basic Security Concepts: Authenticity

Three general ways of proving that you are who you say you are:

- something you know
  - secret handshake, password
  - can (easily) be given to and used by somebody else
- something you have
  - physical items: smart card, RSA token, ...
  - private keys
  - can (easily) be given to and used by somebody else
- something you are
  - physical, physiological or behavioral traits
  - cannot (easily or at all) be given to or used by somebody else
  - cannot (easily or at all) be changed once compromised

Basic Security Concepts: Authenticity

Some possible threats:

- lack of *confidentiality*
- lack of *integrity*
- reliance on fragile infrastructure
- usability
- conflation with *authorization*

Principle of Least Privilege

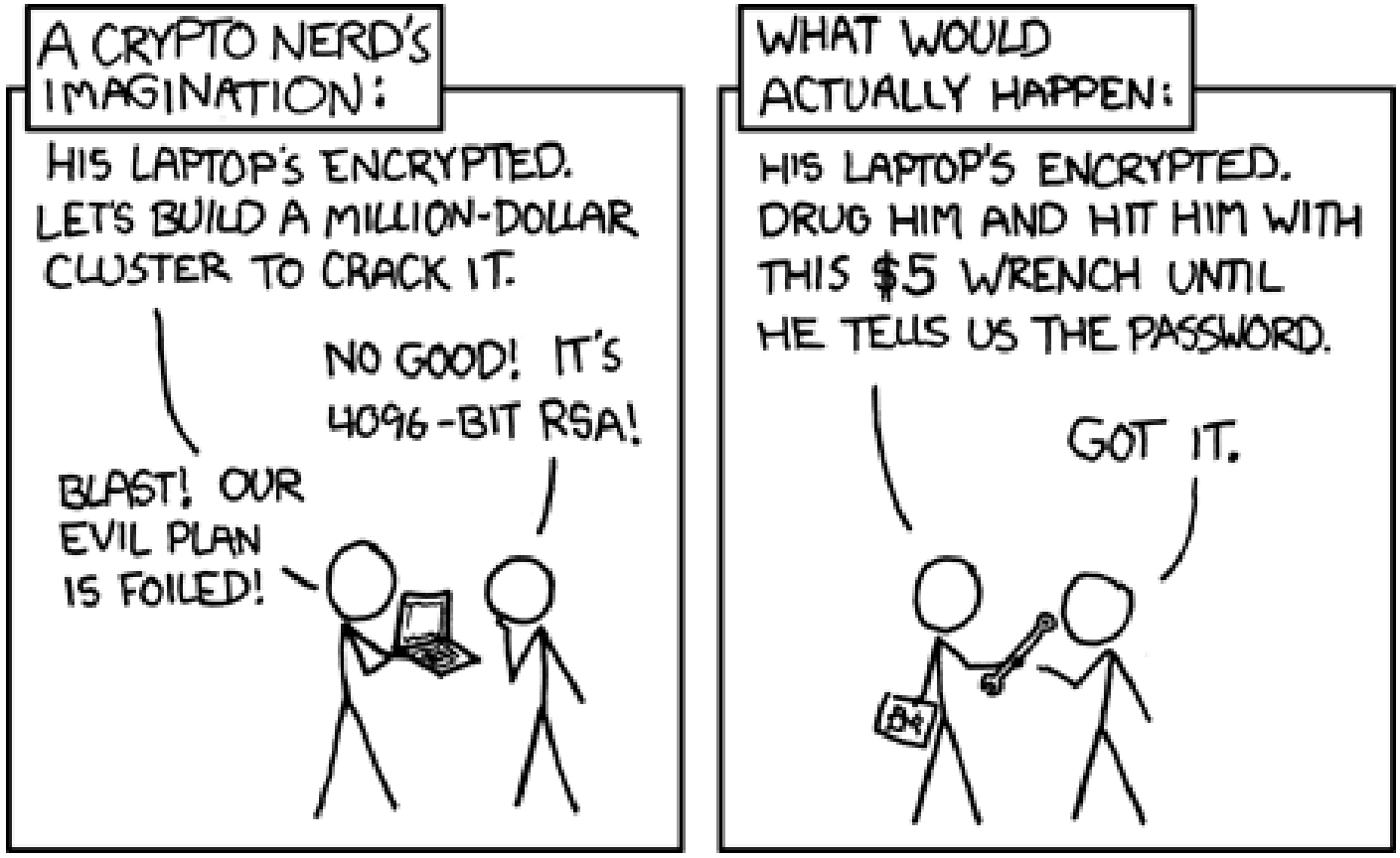
It's not just 1s and 0s

System security is not restricted to *software* security.

It's not just 1s and 0s

The thing that makes security difficult is not the software or hardware components. It's the human component.

It's not just 1s and 0s



Secure by default

Users care about usability, not about security.

Secure by default

Users will not change their default settings.

Secure by default

Users will not change their default settings.

(Unless a less secure option is available.)

Hooray!

5 Minute Break

Security Fallacies and Pitfalls

Proving a Negative

(Evidence of Absences vs. Absence of Evidence)

Security Fallacies and Pitfalls

Security by Obscurity

Security Fallacies and Pitfalls

Perfect is the Enemy of the Good

(Differentiate between futile efforts and raising the bar.)

Security Fallacies and Pitfalls

One in a million is next Tuesday.

`http://is.gd/Isb20K`

Security Fallacies and Pitfalls

"Any person can invent a security system so clever that she or he can't think of how to break it."

Schneier's Law `http://is.gd/hW82dt`

Security Fallacies and Pitfalls

Don’t invent your own crypto.

(Seriously, don’t.)

Security Fallacies and Pitfalls

Complexity is the worst enemy of security.

(The more secure you make something, the less secure it becomes.)

Whom do you trust?

`http://cm.bell-labs.com/who/ken/trust.html`

Outsourcing Services

- you trust the provider/vendor to honor the agreement
- you “hope” they won’t change their agreement (once invested, changing back is hard)
- you trust the provider/vendor to keep their infrastructure safe
- you trust the provider/vendor’s employees
- you are ok with the traffic going across the public internet

Outsourcing Services

- you trust the provider/vendor to honor the agreement
- you “hope” they won’t change their agreement (once invested, changing back is hard)
- you trust the provider/vendor to keep their infrastructure safe
- you trust the provider/vendor’s employees
- you are ok with the traffic going across the public internet

Bottom-line: are you increasing or decreasing your attack surface?

Embrace Automation

Vulnerabilities are dense.

Eliminate *classes* of attacks, not individual flaws.

Build Robust Infrastructures and Service

Your endpoint security model should assume the network is compromised;
your network security model should assume the endpoint is.

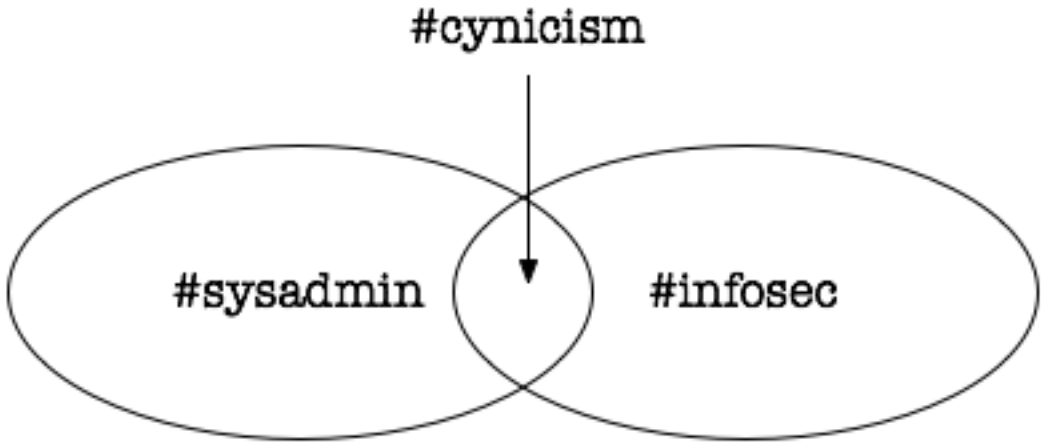
Both in fact are.

Toning down the Paranoia

Never attribute to malice that which can be adequately explained by stupidity.

Hanlon's Razor

Sysadmin ∩ Infosec

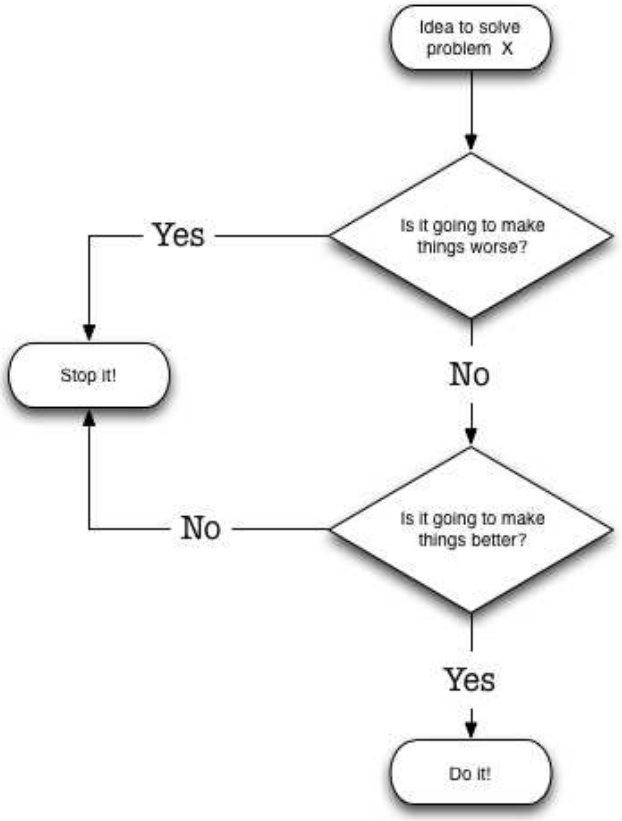


https://www.netmeister.org/blog/infosec-basics.html

Sysadmin ∩ Infosec

Nothing is always absolutely so.

Two Questions



https://www.netmeister.org/blog/two-questions.html

Infosec Foundation

Don’t be lazy.

Final Project

Group project: Capture the Flag

`https://www.cs.stevens.edu/~jschauma/615/ctf.html`

Additional Reading

https://www.slideshare.net/zanelackey/attackdriven-defense
https://www.netmeister.org/blog/moving-the-needle.html
https://twitter.com/jschauma/status/713118376550404096
https://t.co/DRHbEKXod8
https://danielmiessler.com/study/security_and_obscurity/
http://is.gd/sGnRVL