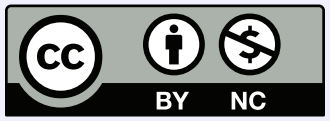
Thinking Security

Steven M. Bellovin
https://www.cs.columbia.edu/~smb



Change

Once in ancient days, the then King of England told Sir Christopher Wren, whose name is yet remembered, that the new Cathedral of St. Paul which he had designed was “awful, pompous and artificial.” Kings have seldom been noted for perspicacity.

. . .

Change

Once in ancient days, the then King of England told Sir Christopher Wren, whose name is yet remembered, that the new Cathedral of St. Paul which he had designed was “awful, pompous and artificial.” Kings have seldom been noted for perspicacity.

...

In the case of the King and Sir Christopher, however, a compliment was intended. A later era would have used the words “awe-inspiring, stately, and ingeniously conceived.

More than Language Changes

- Businesses change
- Threats change
- Technology changes

☞ How can we build secure systems, in a rapidly changing environment?

Businesses and Threats

- What do you want to protect?
- Against whom?

☞ These are the first two questions to ask in any security scenario

Assets

- Different assets require different levels of protection
- Contrast the value of celebrity photos with this very ordinary picture I took
- Security measures have to be commensurate with the value of the assets

Assets and Hackers

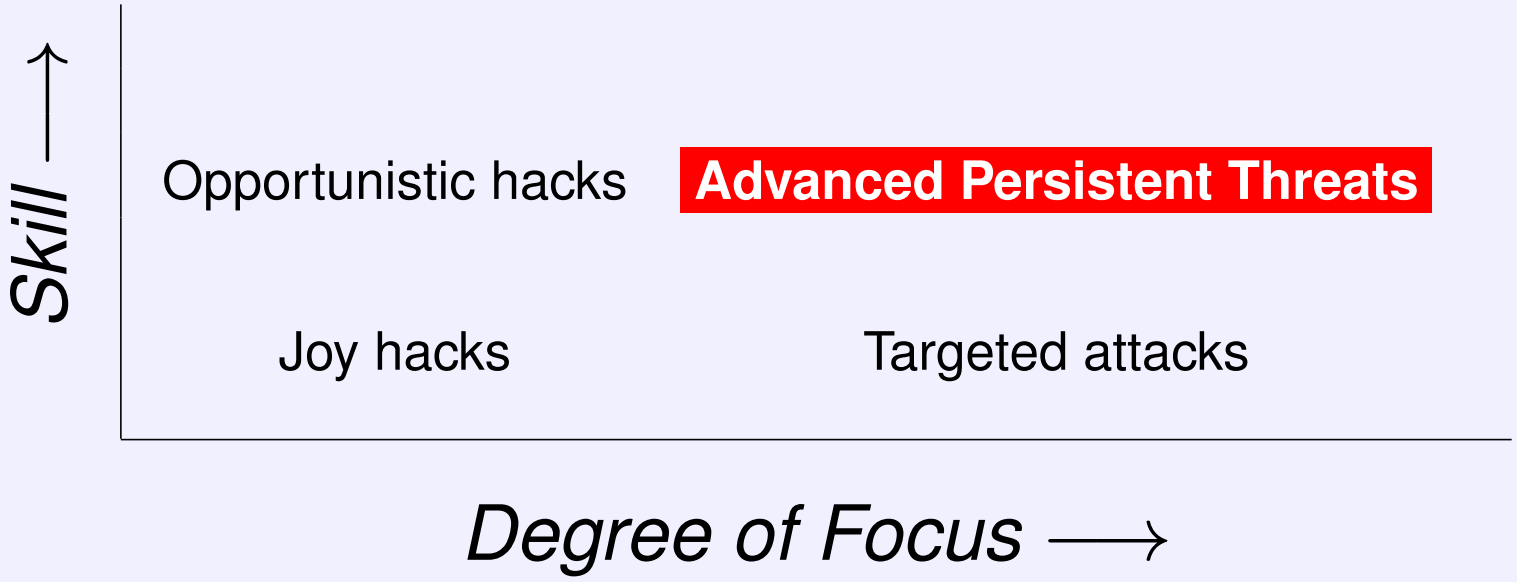
- Different kinds of assets attract different kinds of hackers
- The NSA probably isn't interested in nude celebrity selfies
- (But they may want such pictures if taken by one of their targets.)
- They're very interested in military and political information
- Most hackers, though, want money

☞ They'll go after anything they can monetize

Hackers

- Different kinds of hackers have different skill and different goals
- They also have different degrees of focus—do they really care what they get?

The Threat Matrix



Joy Hackers

- Little skill (mostly runs canned exploit scripts), not very good target selection
- Describes most novices
- Doesn't really care about targets—anyone they can succeed against is whom they were aiming for
- They can do damage, but ordinary care is generally sufficient

Opportunistic Hackers

- Skilled, often very skilled, but they also don't care much about targets
- Most viruses are written by this class of attacker
- Generally speaking, their goal is money: credit cards, bank account credentials, spambots, etc.
- Quite dangerous—but if you're good enough, they'll switch targets

Targetiers

- (An ancient word whose meaning I'm changing. . .)
- Attackers who target you specifically, but aren't that skilled
- Will do in-depth research on their targets, and tailor their attacks accordingly
- May even exploit physical proximity
- Sometimes a disgruntled insider or ex-insider
- Again, quite dangerous

Advanced Persistent Threats

- Very skilled attackers who focus on particular targets
- The best attackers in this class are national intelligence agencies—you know the countries on the list as well as I do. . .
- May discover and employ “0-days”—holes for which no patches exist, because the vendor doesn’t know of the problem
- May employ advanced cryptographic techniques
- Will employ non-computer means of attack as a complement “The Three Bs: burglary, bribery, and blackmail”
- *No* high-assurance defenses

APT

Apt: An Arctic monster. A huge, white-furred creature with six limbs, four of which, short and heavy, carry it over the snow and ice; the other two, which grow forward from its shoulders on either side of its long, powerful neck, terminate in white, hairless hands with which it seizes and holds its prey. Its head and mouth are similar in appearance to those of a hippopotamus, except that from the sides of the lower jawbone two mighty horns curve slightly downward toward the front. Its two huge eyes extend in two vast oval patches from the centre of the top of the cranium down either side of the head to below the roots of the horns, so that these weapons really grow out from the lower part of the eyes, which are composed of several thousand ocelli each. Each ocellus is furnished with its own lid, and the apt can, at will, close as many of the facets of his huge eyes as he chooses.

Edgar Rice Burroughs, *Thuvia, Maid of Mars*

Who are the APTs?

- The US blames China and Russia
- China blames the US. (So do most other countries. . .)
- Iran blames Israel
- Israel blames Iran and Iranian-backed Palestinians
- Etc.

- I'll blame beings from the Andromeda galaxy—this way, I don't have take sides

Assessing Risk

- What assets do you have?
- What classes of attackers would be interested in them?
- How powerful are those attackers?
- How much security should you afford?

Business and (In)Security

- The purpose of a business (or other organization, but for simplicity I'll speak of businesses) is *not* to stay secure
- Rather, it's to achieve certain goals
- From that perspective, insecurity is simply a cost, *not a state of sin*
- So are security measures. . .
- What is the right tradeoff?

Insecurity

- I'll repeat that: insecurity is not a state of sin
- It is perfectly reasonable to omit certain security measures if their cost is too high relative to the threats you face
- However—be very, very certain that you understand the assets at risk and who might go after them

Target Selection

- The attackers have gotten quite sophisticated at target selection
- They've gone after little-known sectors like credit card payment processors
- Governments often want to build up their own industries, which means that industrial secrets of any sort are at risk from APTs
- Passwords from otherwise-uninteresting sites may be valuable because people tend to reuse passwords elsewhere, including on financial sites
- Don't forget your company's legacy systems

Case Study: Manning and the Wikileaks Cables

- Much of the US government has come to believe that too much compartmentalization was bad, and loosened access controls on some information
- Their defenses against external attackers were pretty good
- They thought there were no insider risks
- Result: Manning downloaded ~250,000 “cables” and leaked them

Case Study: Mobile Phone Cloning

- Early US mobile phones were easily cloned: an eavesdropper could pick up ESN/MIN pairs over the air, and burn one into another phone
- The designers had realized this, but overestimated the cost of the attack, the skill level required, and the distribution of such skills

☞ Electronics repair technicians simply bought off-the-shelf test gear

- They assumed limited use of mobile phones (not many targets) and a motive of cost-avoidance
- In fact, phones became widespread, and the motive was criminals wishing to avoid wiretaps

☞ The attack was easier and the attackers had stronger motives than had been anticipated

Case Study: The Crazy Neighbor Attack

- A family angered a neighbor by (justifiably) calling the police about his behavior
- He spent weeks cracking their WiFi password, hacking their computers, and attempting to frame them for various crimes, including child ponrography, sexual harassment, and threatening the Vice President
- The family's defenses assumed opportunistic attackers, but they were targeted

Case Study: Stuxnet

- The Iranians assumed that their uranium centrifuge plant was being targeted by serious adversaries—and of course they were right
- They thought that an air gap would defend the plant's network
- The attackers were more powerful than they had assumed

Assumptions

- Why should technology changes affect our security reasoning?
- Speed? Applications? Bandwidth?
- Many of our security architectures are built around *implicit assumptions*—and since we don't know what they are, we don't react when they're violated
- We have to identify those assumptions

Example: Passwords

- Assumption: attacker's computational power is a very small number of computers

☞ Today, they have botnets with GPUs

☞ Result: guessing attacks are far more effective

- Assumption: users are primarily employees, who could be trained

☞ Today, it's mostly users who will shop or bank elsewhere if they don't like a site's rules

☞ That's why popular passwords include "123456", "12345", "password", "iloveyou", etc.

- More examples tomorrow

Example Assumptions: Smartphones

- Assumption: the IT department controls all devices

☞ Smartphones are often employee-owned and operated devices, on *your* network

☞ They're also on home networks, hotel network, mobile phone networks, and more

We Won't Identify All Implicit Assumptions

- We can't—by definition, they're *implicit*
- We can try asking, in different places, "why do you think this is secure?"
- In addition, deployed architectures should be reviewed every few years, to ensure that it is still sound and to exam unreviewed changes to the architecture

Thinking About Insecurity

- In order to know how to defend systems, you have to know how to attack them
- What sorts of attacks are launched?
- Why do they sometimes succeed when you did get the threat model correct?

Thinking Sideways

- Attacks frequently succeed when the attacker thinks of an input pattern that the programmer didn't anticipate
- If the choices for an exam question are (a), (b), or (c), enter (d)
- If you that doesn't work, can you sabotage the test?
- "You don't go through strong security, you go around it"

The *Kobayashi Maru*

- In a *Star Trek* movie, cadets were presented with a no-win situation and asked to solve it
- Kirk snuck in and reprogrammed the simulation computer to make a solution possible

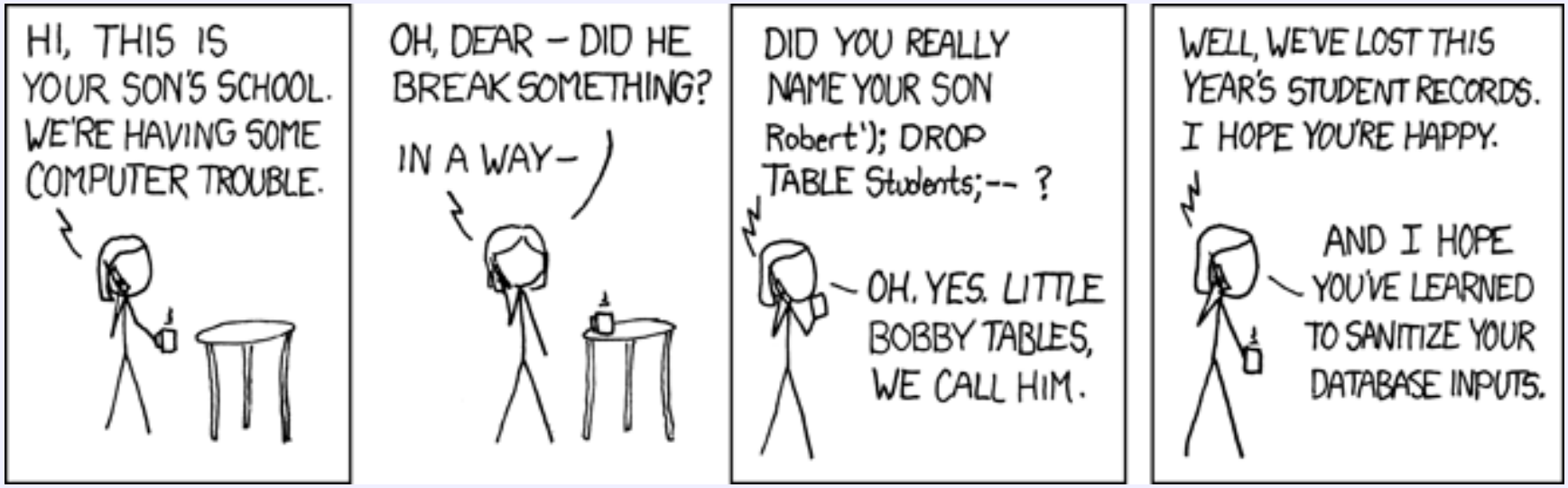
Bruce Schneier on Uncle Minton's Ant Farms

- You can buy an ant habitat as a child's toy
- It comes with a card you mail in to get a tube of ants
- Friend: I didn't know you could send ants through the mail
- Bruce: Gee, I can get them to send a tube of ants to *anyone*

Attackers Don't Follow the Rules

- Requirements document: "Program must accept input lines of 1024 characters"
- Programmer: "char buf[1025]; // leave room for NUL byte"
- Tester: "It accepted the 1024-byte test line; requirement fulfilled"
- Hacker: "What happens if I send 2000 bytes?"

Little Bobby Tables



(http://xkcd.com/327/)

And in Real Life

As you may have heard, we've had a very close election here in Sweden. Today the Swedish Election Authority published the hand written votes. While scanning through them I happened to notice

```
R;13;Hallands län;80;Halmstad;01;Halmstads västra
valkrets;0904;Söndrum 4;pwn DROP TABLE VALJ;1
```

The second to last field[1] is the actual text on the ballot[2]. Could it be that Little Bobby Tables is all grown up and has migrated to Sweden? Well, it's probably just a joke but even so it brings questions since an SQL-injection on election data would be very serious.

Someone even tried to get some JavaScript in there:

(`http://alicebobandmallory.com/articles/2010/09/23/did-little-bobby-tables-migrate-to-sweden`)

Security is a Systems Problem

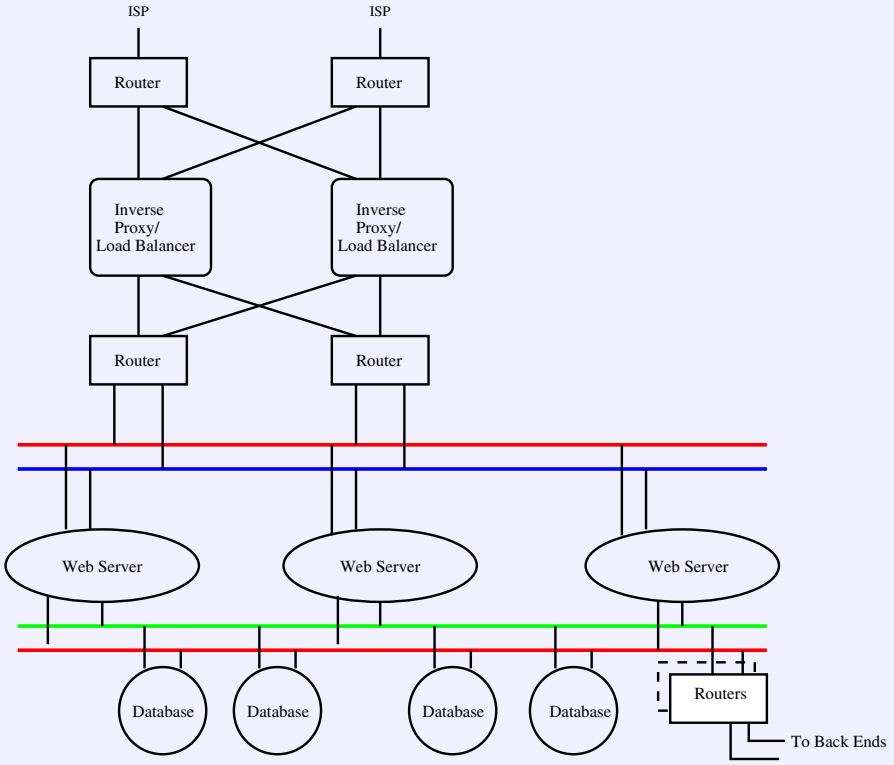
- You don't get security by sprinkling on crypto
- You don't get security by requiring strong passwords
- You don't get security by adding firewalls
- All of these help—but components interact, and it's often the interactions that cause the problems
- Example: if you encrypt a file, you move the insecurity from the file's storage to the key's storage—and you risk losing the file if you lose the key

Breaking Web Cryptography

- Suppose you want to read encrypted web traffic
- You can: (a) break RSA; (b) break RC4 (allegedly, the NSA can) or AES; (c) hack a certificate authority and issue yourself a fake cert for that site; (d) find a flaw in the SSL implementation and use it to recover the private key (Heartbleed); (e) hack the web server or the client systems to send you the plaintext; (f) bribe a server site employee to plant a back door for you; (g) etc.

A Server Farm

- In a typical web server complex, the inverse proxies act as a firewall, allowing only ports 80 and 443 through
- You can't get at the databases from the Internet unless you first hack the web servers
- But what about that link at the lower right to the rest of the company?



Evaluating System Designs

- How do we avoid these traps?
- Draw the system diagram
- For each node and each link:
  - Assume that it has been compromised
  - Assess the odds of this happening
  - What are the consequences?

  For each serious situation, where the odds are high and the consequences serious, find a defense

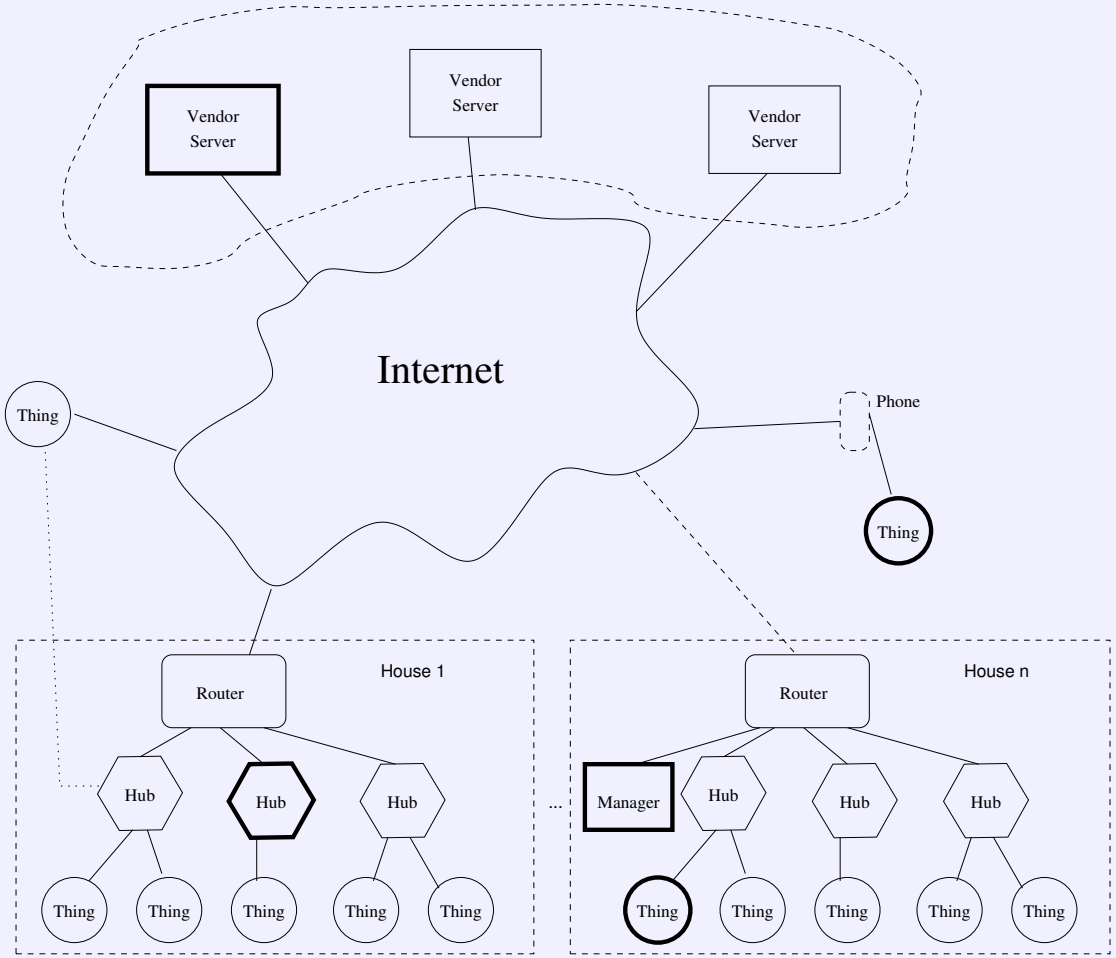
Is a Given Subsystem Secure?

- We can't really tell
- We can use heuristics, including historical record (some software packages are notoriously insecure)
- Complexity is bad: complex code is buggy more often, and buggy code is often insecure
- We can compare two alternatives using *Relative Attack Surface Quotient* (RASQ)

Relative Attack Surface Quotient

- A comparative (*not* absolute) measure of how many ways there are to attack different systems
- Count up vulnerable points: open sockets, privileged programs, loose access control restrictions, etc.
- Weight each one
- Compare the total for different alternatives

The Internet of Things



Explanation

- “Things” talk to their hubs
- Hubs talk to Internet-resident vendor servers
- Vendor servers talk to each other
- Because of NATs, etc., little direct communication to the Things; control and data go via the servers

Failure Types

- Links—but encryption is (relatively) easy
- Servers hacked
- Hubs hacked
- Devices hacked

Servers Hacked

- Hard to protect—they're Internet-resident, and have to listen to many places
- A hacked server can send bad commands or bad firmware to Things, and authentication data is at risk
- Defense: firmware must be digitally signed, by offline machines
- (The Andromedans can hack that, but they're not in the threat model)
- Defense: good (preferably hardware) sanity checking on commands
- Defense: don't use passwords; use asymmetric crypto
- Defense: authorization must be done by Things or hubs, not servers

Hubs

- Not directly expose to Internet, hence harder to attack
- Hubs are mostly message relays, so there's little new risk
- Defense: intrusion detection
- Thing-to-Thing messages must be authenticated end-to-end, but encrypted hop-by-hop, to permit intrusion detection.
- New added risk: we've just complicated the key management

Things

- Hacked things can report bad data, and can do nasty stuff to the associated device
- Defense: sanity-checking by the hubs and servers; hardware safety limits to prevent dangerous outcomes

The Humans

- Normal people will have to control the crypto, the authentication, and the authorization lists
- Automate most of that (like key management) out of existence
- New added risk: fancy crypto is dangerous crypto
- Pay a *lot* of attention to the human interface for authorization
- ☞ The literature is quite clear: standard ACLs are impossible for most people to manage

Overall Risk Ranking

- Server compromise is the most serious risk, because they're exposed and because they control so many Things
- Usability errors (including home PCs being infected by spam emails with clickable nasties) are probably the second-biggest

Today's Status

- Too many Internet of Things links aren't encrypted—and that's the easy defense
- Few companies think about usable security
- We're probably in trouble. . .

What are our Assumptions?

- Vendors are trustworthy, and don't try to subvert security
- Active attacks are hard
- Houses are isolated from the Internet by NAT boxes, which (partially) act as firewalls
- Cryptography implementable on Things is secure
- The Andromedans aren't the enemy

Let's Look at the Last One

- Andromedans can hack the vendor's firmware-signing machine: no defense against malicious code
- Andromedans can break into your house and get on your network that way
- They can even tamper with the design of the failsafe hardware limits

Conclusions

- Analyze the risks: what are you protecting, and against whom?
- How powerful are your adversaries?
- Who can take out which elements of your systems?
- How can you stop them?