GRAD SEC

A WHIRLWIND TOUR

CMSC 818O

AUG 31 2017

TODAY'S PAPERS

Schneier on Security

Blog | Newsletter | Books | Essays | News | Talks | Academic | About Me

Blog >

The Security Mindset

Uncle Milton Industries has been selling ant farms to children since 1956. Some years ago, I remember opening one up with a friend. There were no actual ants included in the box. Instead, there was a card that you filled in with your address, and the company would mail you some ants. My friend expressed surprise that you could get ants sent to you in the mail.

I replied: "What's really interesting is that these people will send a tube of live ants to anyone you tell them to."

Security requires a particular mindset. Security professionals -- at least the good ones -- see the world differently. They can't walk into a store without noticing how they might shoplift. They can't use a computer without wondering about the security vulnerabilities. They can't vote without trying to figure out how to vote twice. They just can't help it.

SmartWater is a liquid with a unique identifier linked to a particular owner. "The idea is for me to paint this stuff on my valuables as proof of ownership," I wrote when I first learned about the idea. "I think a better idea would be for me to paint it on your valuables, and then call the police."

Really, we can't help it.

This kind of thinking is not natural for most people. It's not natural for engineers. Good engineering involves thinking about how things can be made to work; the security mindset involves thinking about how things can be made to fail. It involves thinking like an attacker, an adversary or a criminal. You don't have to exploit the vulnerabilities you find, but if you don't see the world that way, you'll never notice most security problems.

I've often speculated about how much of this is innate, and how much is teachable. In general, I think it's a particular way of looking at the world, and that it's far easier to teach someone domain expertise -- cryptography or software security or safecracking or document forgery -- than it is to teach someone a security mindset.

Which is why CSE 484, an undergraduate computer-security course taught this quarter at the University of Washington, is so interesting to watch. Professor Tadayoshi Kohno is trying to teach a security mindset.

You can see the results in the blog the students are keeping. They're encouraged to post security reviews about random things: smart pill boxes, Quiet Care Elder Care monitors, Apple's Time Capsule, GM's OnStar, traffic lights, safe deposit boxes, and dorm room security.

One recent one is about an automobile dealership. The poster described how she was able to retrieve her car after service just by giving the attendant her last name. Now any normal car owner would be happy about how easy it was to get her car back, but someone with a security mindset immediately thinks: "Can I really get a car just by knowing the last name of someone whose car is being serviced?"

The rest of the blog post speculates on how someone could steal a car by exploiting this security vulnerability, and whether it makes sense for the dealership to have this lax security. You can quibble with the analysis -- I'm curious about the liability that the dealership has, and whether their insurance

Search

Powered by DuckDuckGo

Go

blog essays whole site

Subscribe

About Bruce Schneier

I've been writing about security issues on my blog since 2004, and in my monthly newsletter since 1998. I write books, articles, and academic papers. Currently, I'm the Chief Technology Officer of IBM Resilient, a fellow at Harvard's Berkman Center, and a board member of EFF.

Related Entries

- Nice Security Mindset Example
- Teaching the Security Mindset
- What is a Hacker?
- IT Security and the Normalization of Deviance
- NSA Cryptography Course

Featured Essays

- The Value of Encryption
- Data Is a Toxic Asset, So Why Not Throw It Out?
- How the NSA Threatens National

Why Information Security is Hard
– An Economic Perspective

Ross Anderson

University of Cambridge Computer Laboratory,
JJ Thomson Avenue, Cambridge CB3 0FD, UK
Ross.Anderson@cl.cam.ac.uk

Abstract

According to one common view, information security comes down to technical measures. Given better access control policy models, formal proofs of cryptographic protocols, approved firewalls, better ways of detecting intrusions and malicious code, and better tools for system evaluation and assurance, the problems can be solved.

In this note, I put forward a contrary view: information insecurity is at least as much due to perverse incentives. Many of the problems can be explained more clearly and convincingly using the language of microeconomics: network externalities, asymmetric information, moral hazard, adverse selection, liability dumping and the tragedy of the commons.

1 Introduction

In a survey of fraud against automatic teller machines [4], it was found that patterns of fraud depended on who was liable for them. In the USA, if a customer disputed a transaction, the onus was on the bank to prove that the customer was mistaken or lying; this gave US banks a motive to protect their systems properly. But in Britain, Norway and the Netherlands, the burden of proof lay on the customer: the bank was right unless the customer could prove it wrong. Since this was almost impossible, the banks in these countries became careless. Eventually, epidemics of fraud demolished their complacency. US banks, meanwhile, suffered much less fraud; although they actually spent less money on security than their European counterparts, they spent it more effectively [4].

There are many other examples. Medical payment systems that are paid for by insurers rather than by hospitals fail to protect patient privacy whenever this conflicts with the insurer's wish to collect information about its clients. Digital signature laws transfer the risk of forged signatures from the bank that relies on the signature (and that built the system) to the person alleged to have made the signature. Common Criteria evaluations are not made by the relying party, as Orange Book evaluations were, but by a commercial facility paid by the vendor. In general, where the party who is in a position to protect a system is not the party who would suffer the results of security failure, then problems may be expected.

A different kind of incentive failure surfaced in early 2000, with distributed denial of service attacks against a number of high-profile web sites. These exploit a number of subverted machines to launch a large coordinated packet flood at a target. Since many of them flood the victim at the same time, the traffic is more than the target can cope with, and because it comes from many different sources, it can be very difficult to stop [7]. Varian pointed out that this was also a case of incentive failure [20]. While individual computer users might be happy to spend $100 on anti-virus software to protect themselves against attack, they are unlikely to spend even $1 on software to prevent their machines being used to attack Amazon or Microsoft.

This is an example of what economists refer to as the 'Tragedy of the Commons' [15]. If a hundred peasants graze their sheep on the village common, then whenever another sheep is added its owner gets almost the full benefit – while the other ninety-nine suffer only a small decline in the quality of the grazing. So they aren't motivated to object, but rather to add another sheep of their own and get as much of the grazing as they can. The result is a dustbowl; and the solution is regulatory rather than technical. A typical tenth-century Saxon village had community mechanisms to deal with this problem; the world of computer security still doesn't. Varian's proposal is that the costs of distributed denial-of-service attacks should fall on the operators of the networks from which the flood

THE SECURITY MINDSET

To anticipate attackers
we must be able to think like attackers

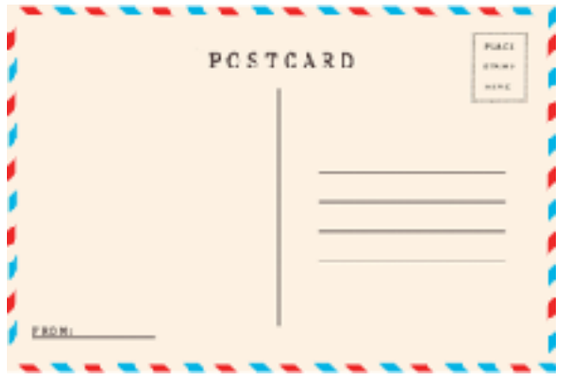
Uniquely identifiable liquid

What would an attacker do?

Paint it on *someone else's* property
and then call the cops

THE SECURITY MINDSET

To anticipate attackers
we must be able to think like attackers



Fill out a card with your address

⇒

They deliver a box of live ants to you

What would an attacker do?

Order them to *someone else*

THE SECURITY MINDSET

The ability to view a large, complex system and be able to reason about:

- What are the potential security threats?
- What are the hidden assumptions?
- Are the *explicit* assumptions true?
- How can we mitigate the risks of the system?

Be creative! (Attackers will be)

WHAT DOES IT MEAN TO BE SECURE?

There is no such thing as security,
only degrees of insecurity.

Goal: Raise the bar for the attacker

- Too difficult
- Too expensive
- Lower ROI than the next target

Ultimately, we want to mitigate undesired behavior

WHAT ARE "UNDESIRED" BEHAVIORS?

- Reveals info users wish to hide (confidentiality)
  - Corporate secrets
  - Private data; personally identifying information (PII)

- Modifies information or functionality (integrity)
  - Destroys records
  - Changes data in-flight (think "the telephone game")
  - Installs unwanted software (spambot, spyware, etc.)

- Denies access to a service (availability)
  - Crashing a website for political reasons
  - Denial of service attack
  - Variant: fairness

This is a subset

ATTACKS ARE COMMON

WHY ARE ATTACKS COMMON?

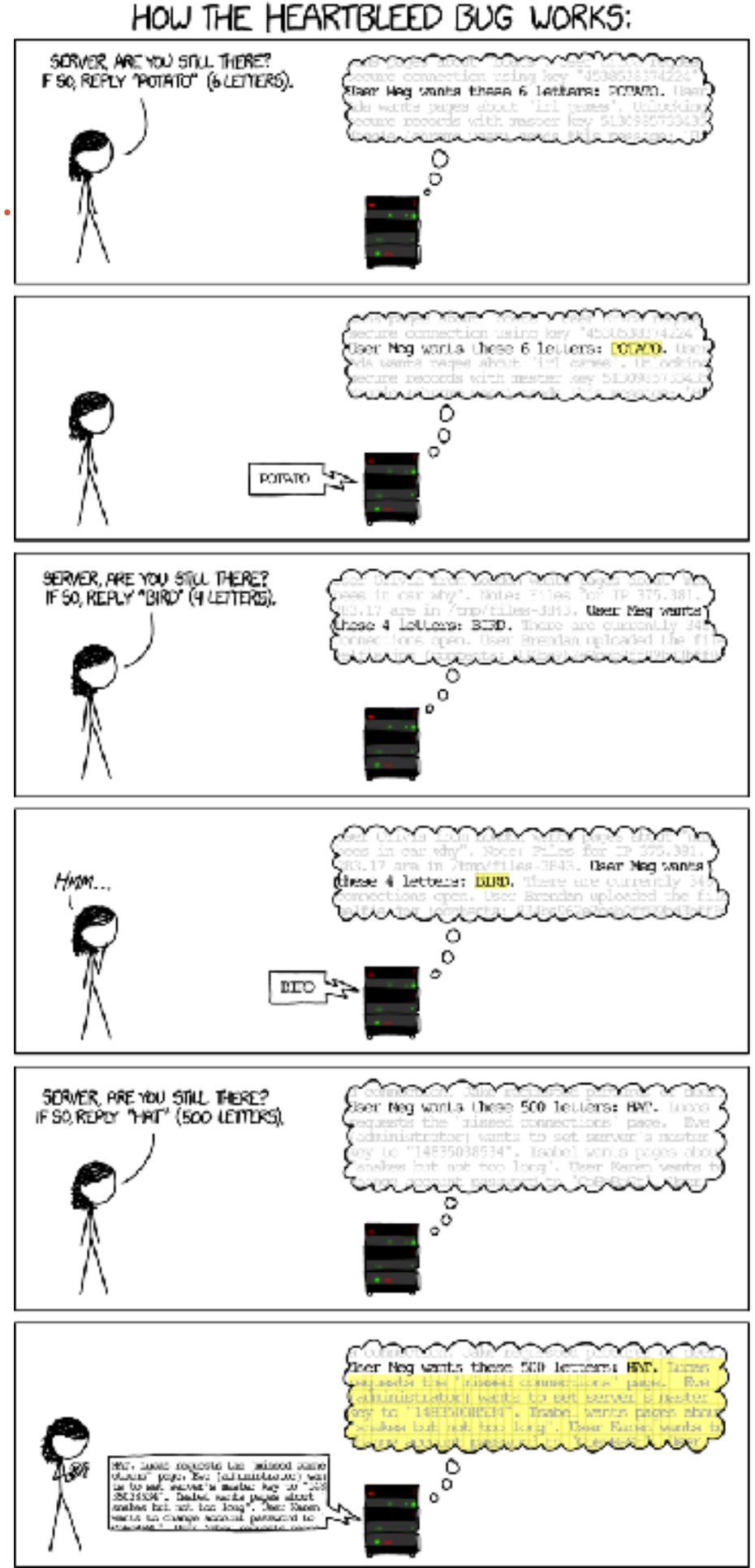
- Security is a property of the systems *we build*

- Many attacks begin by exploiting a vulnerability
  - Vulnerability = defect in hw, sw, protocol, design, … that can be exploited to yield an undesired behavior
  - Software defect = the code doesn't "behave correctly"

- Defects arise due to
  - flaws in the design and/or
  - bugs in the implementation

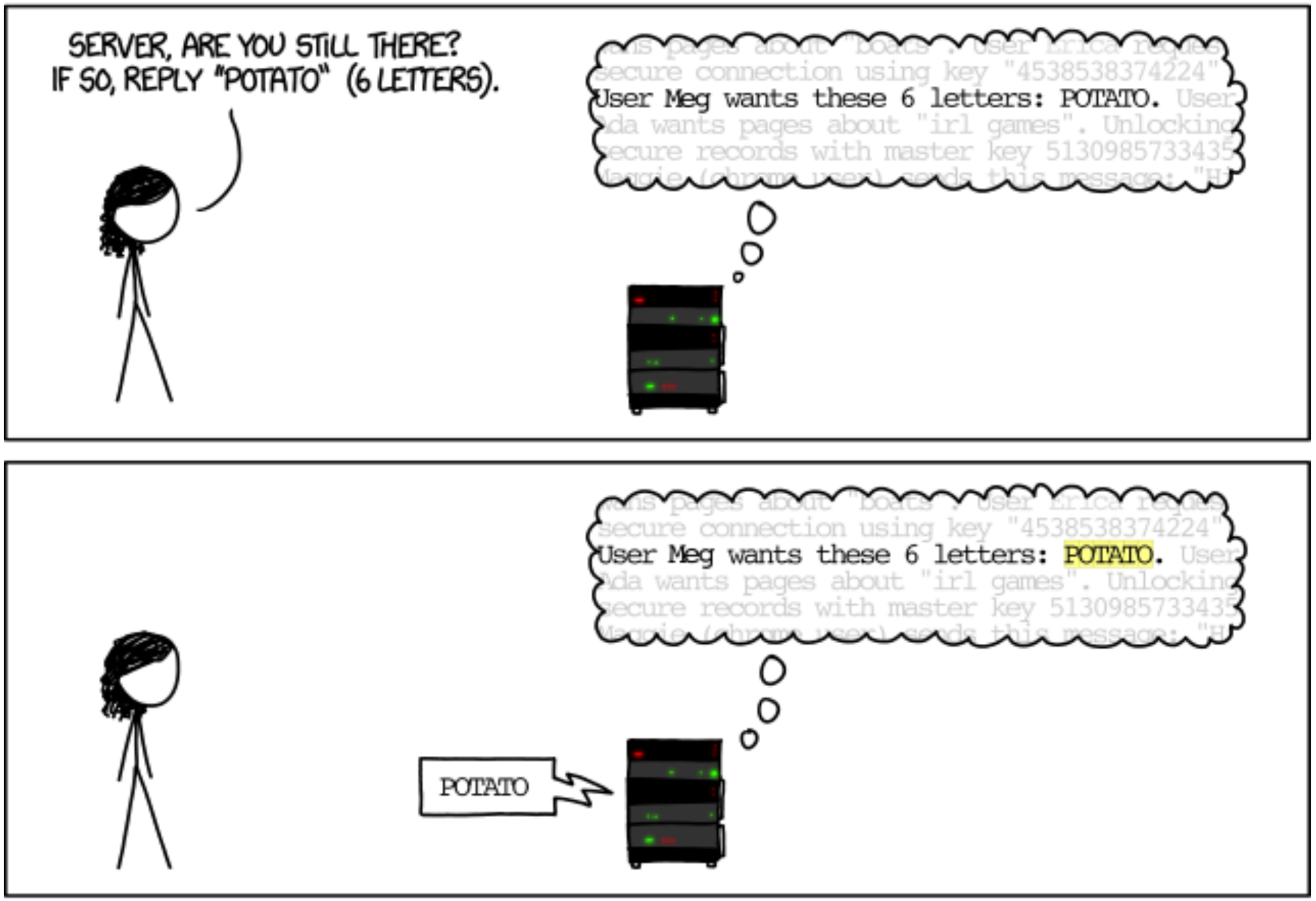
HEARTBLEED

- SSL is the de facto protocol for secure online communication
- Heartbleed was a vulnerability in the most popular SSL server
  - A malformed packet allows you to see server memory
- Fix: don't let the user just tell you how much data to give back
- This was a design flaw

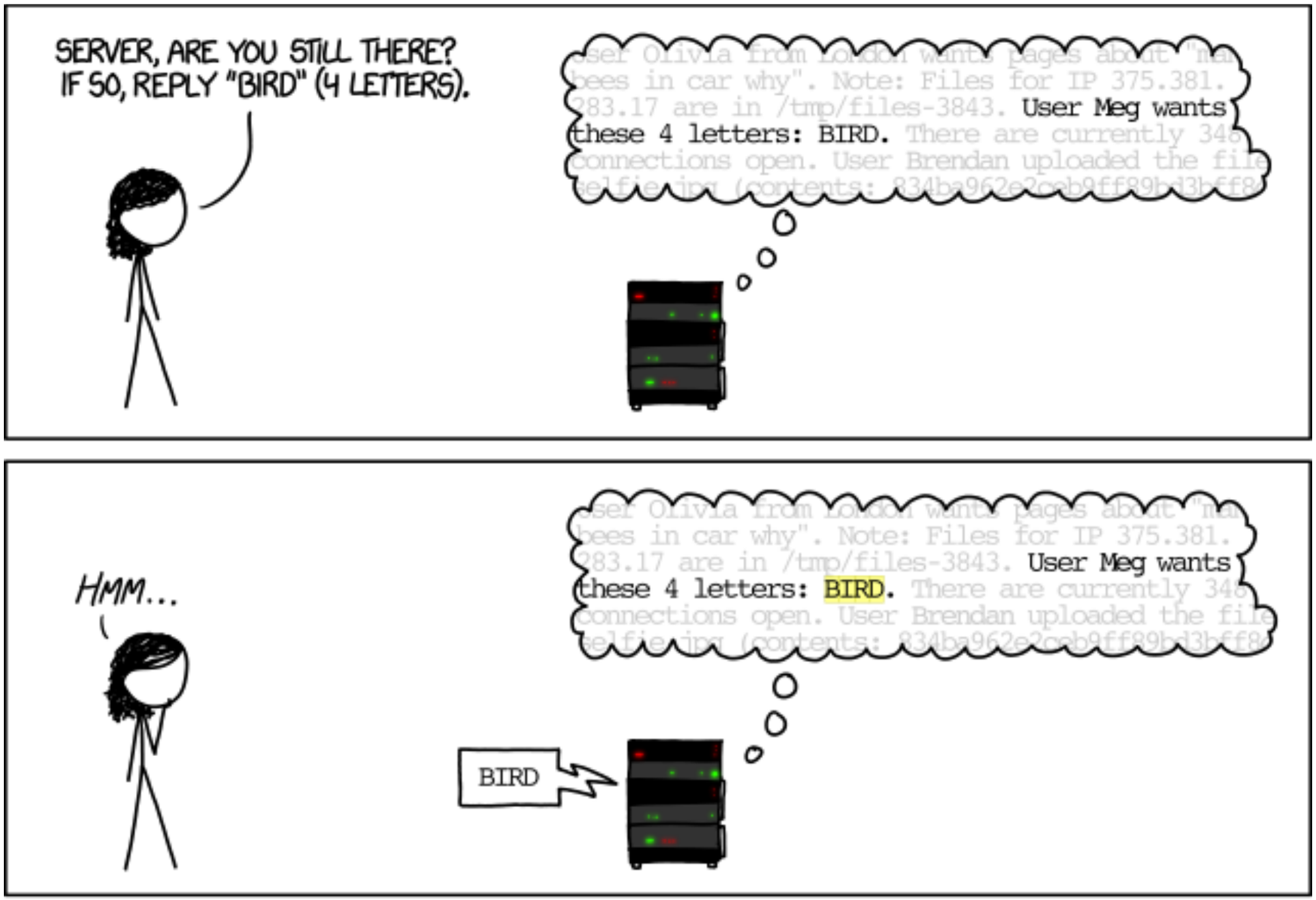


HEARTBLEED

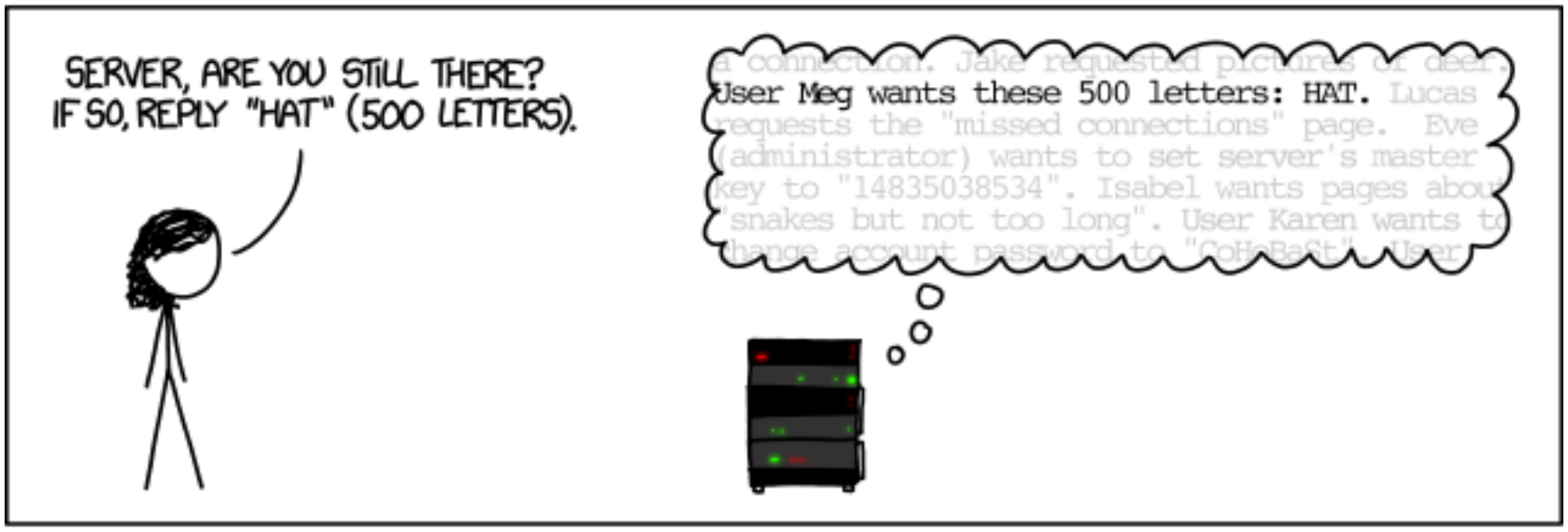
HOW THE HEARTBLEED BUG WORKS:



HEARTBLEED


SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
User Olivia from London wants pages about "ma bees in car why". Note: Files for IP 375.381. 283.17 are in /tmp/files-3843. User Meg wants these 4 letters: BIRD. There are currently 346 connections open. User Brendan uploaded the file selfie.jpg (contents: 834ba962e2ceb9ff89bd3bff8
HMM...
BIRD

HEARTBLEED



User passwords, private keys, personal information…

~40% of “secure” web servers vulnerable

RSA 2011 BREACH

1. **Carefully crafted Flash program**. When run by the vulnerable Flash player, allows the attacker to execute arbitrary code on the running machine.

2. This program could be **embedded in an Excel spreadsheet**, and run automatically when the spreadsheet was opened.

3. Spreadsheet **attached to an email**, masquerading as a trusted party (“spearphishing”)
   - You can forge any “From” address

WHY ARE ATTACKS COMMON?

- Because attacks derive from design flaws or implementation bugs

- But all software has bugs: so what?

- A *normal user* never sees most bugs
  - Post-deployment bugs are usually rare corner cases

- Too expensive to fix every bug
  - Only fix what's likely to affect normal users

WHY ARE ATTACKS COMMON?

Attackers are not normal users

- Normal users avoid bugs/flaws
- Adversaries seek them out and try to *exploit* them

This extends beyond software:

Attacks are possible even with perfect software

WHY ARE ATTACKS COMMON?

Because it's **profitable**

And because a system is
*only as secure as its **weakest link***

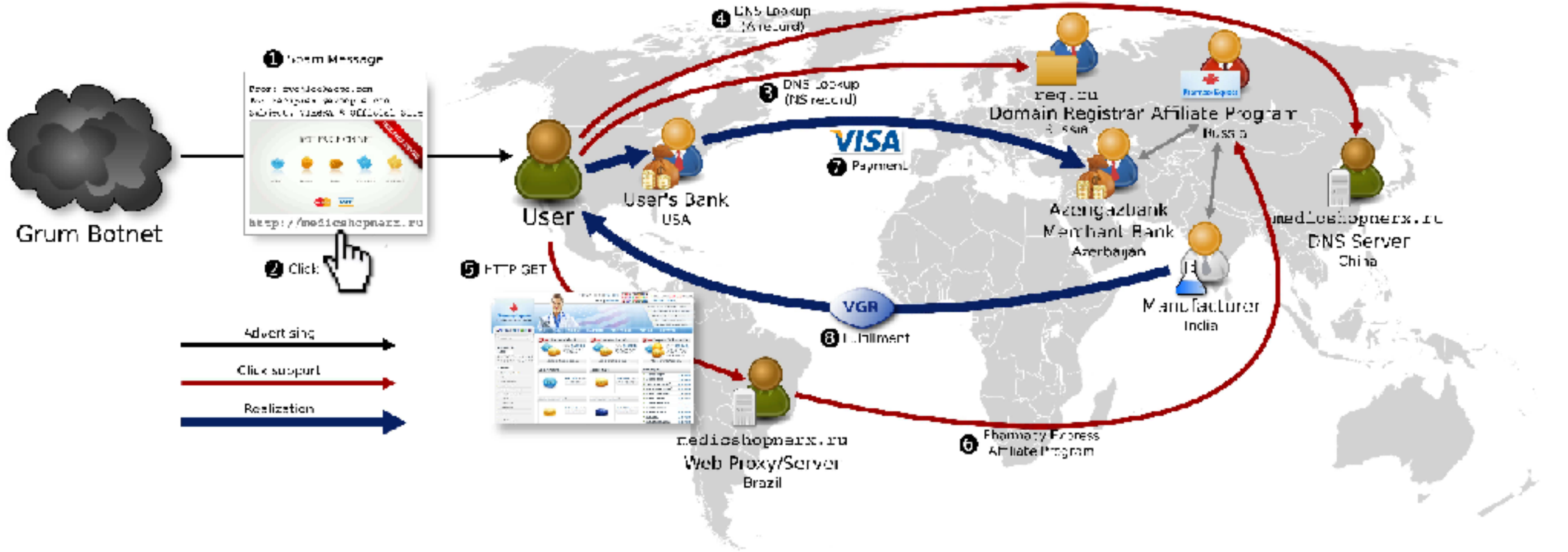


Figure 1: Infrastructure involved in a single URL's value chain, including advertisement, click support and realization steps.

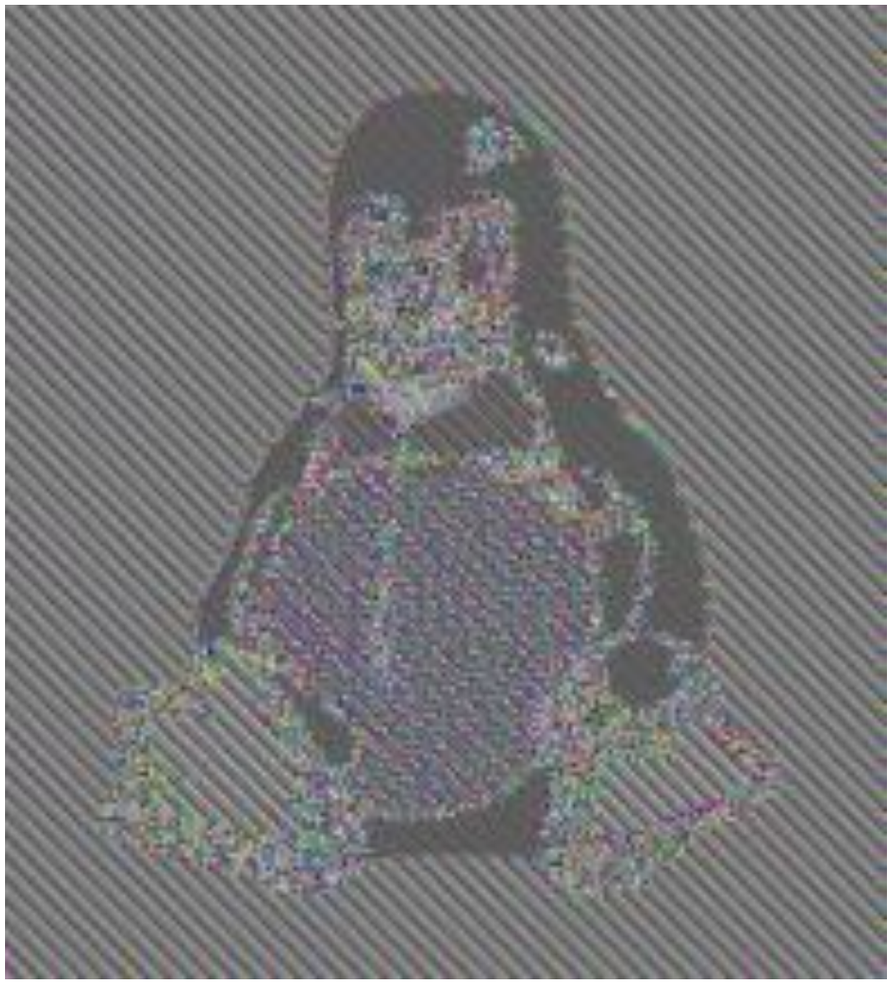
In order to achieve security, we must:

Be able to eliminate bugs and design flaws
and/or make them harder to exploit.

Be able to think like attackers.

Develop a foundation for deeply understanding
the systems we use and build.

UNDERSTANDING THE SYSTEMS WE USE



This is an encrypted image

50% of Android apps that use crypto encrypt in this manner

UNDERSTANDING THE SYSTEMS WE USE

Three things *all* vulnerable websites should have done:

Patch software	Reissue new certificate	Revoke old certificate
93%	27%	13%

Long expiration times:

We will be dealing with Heartbleed for *years*

WHY IS SECURITY DIFFICULT?

Security is indeed a matter of technical reasons.

But “insecurity is at least as much due to perverse incentives”

27% Reissue new certificate

13% Revoke old certificate

Some certificate authorities give certificates for free but *charge to revoke*

TOPICS
OF THIS CLASS

ETHICS IN SECURITY RESEARCH

QUESTION How do we perform research such that the benefit to society outweighs the risk?

PAPERS “Encore” and “All your contacts…”

MEMORY SAFETY

QUESTION How can we safely store and process user input?

ATTACKS Software stores user input in memory.
The attacker exploits this to inject code, exfiltrate data, etc.

DEFENSES **Detect** disallowed memory reads/writes
Taint tracking to find unintended info leakage

PAPERS

Smashing the stack
Flesh on the bone
EXE

Stackguard
Taint tracking
CFI

WEB SECURITY

QUESTION How can we protect users from malicious websites & malicious users on benign websites?

ATTACKS Upload malicious data (XSS, CSRF, SQL injection)
Attack visual integrity (clickjacking)

DEFENSES Secure state shared between site & user (cookies)
Add protections at large hosting providers (CDNs)

PAPERS

SQL Injection
Clickjacking

Defenses for CSRF
Secure delivery networks

USABLE SECURITY

QUESTION How do we properly account for humans?
What can we expect them (not) to do?

ATTACKS password, 123456 (sigh)
Spearphishing, bad interfaces

DEFENSES Improve understanding of user abilities/limitations
Better interfaces and detection of attacks

PAPERS

Password reuse

Spearphishing

Users are not the enemy

Why Johnny can’t encrypt

ISOLATION

QUESTION How can we safely share computing resources between benign and malicious users?

ATTACKS Side-channel attacks
Rowhammer (exploits hardware feature)

DEFENSES Close side-channels
Sandboxes

PAPERS "Get off my cloud" Native Client
Rowhammer Chromium browser

MALWARE

QUESTION How can we detect and mitigate malicious software? What does it do? Who does it?

ATTACKS Viruses, worms, botnets. Various attack vectors (how it infects) and payloads (what it does)

DEFENSES Detection of malware through signatures, metadata, and driveby download nets

PAPERS

Hunting for metamorphic

Ghost in the browser

Inside Slammer

How to 0wn the internet

UNDERGROUND ECONOMIES

QUESTION Who is actually launching these attacks? What are the weak points in these economies?

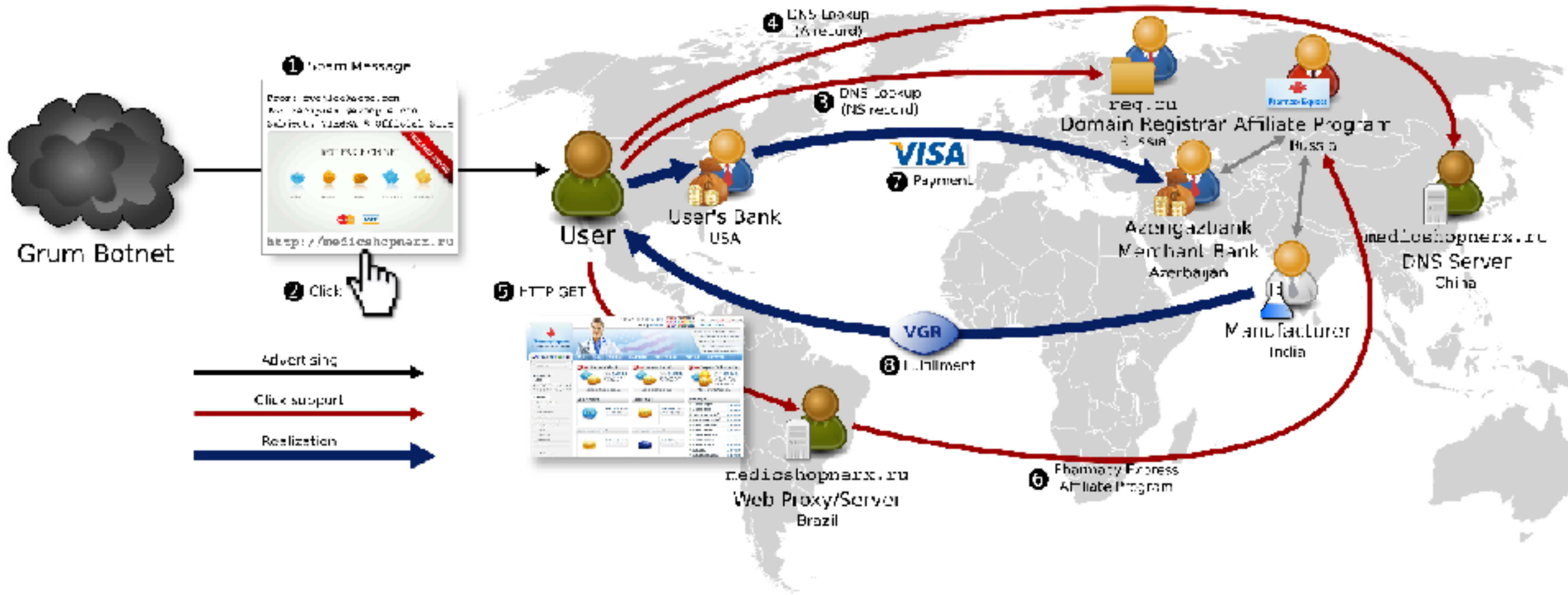


Figure 1: Infrastructure involved in a single URL's value chain, including advertisement, click support and realization steps.

PAPERS Click trajectories Show me the money

CRYPTO FOUNDATION

GOAL

A black-box approach: this is not a crypto class
How to use it properly, how TLS works

QUESTIONS

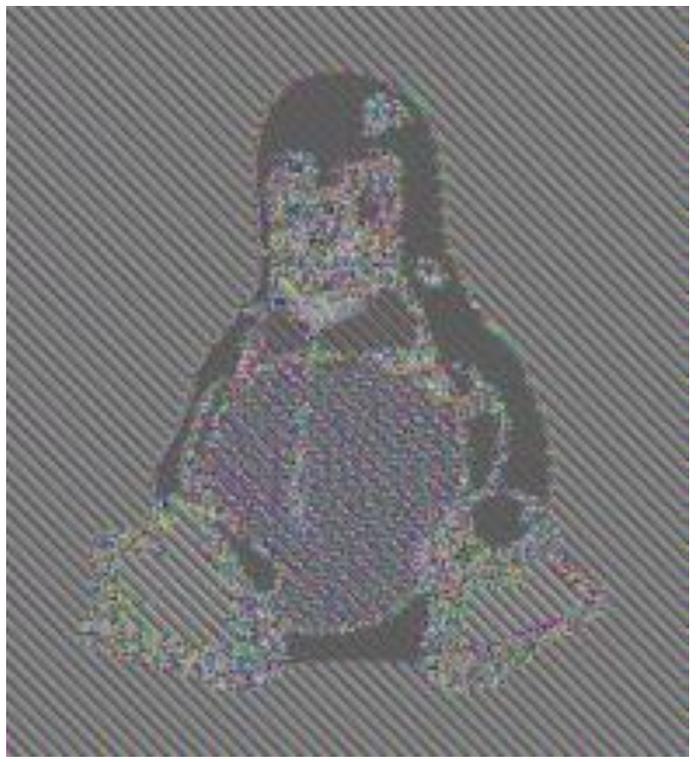
Why does crypto fail in practice?
How do we use these building blocks to build more complicated systems?

PAPERS

TLS/SSL — Diffie-Hellman atk
HTTPS — Most dangerous code…

MEASURING CRYPTO USE IN PRACTICE

QUESTION How is crypto being misapplied or mismanaged?



PAPERS

Measurements of the cert ecosystem

Crypto misuse in Android apps

NEW CRYPTO MECHANISMS

GOAL Understand how to apply cryptographic techniques to build new systems

MECHANISMS Property-preserving encryption
Group signatures
Blockchains

PAPERS CryptDB Attacking CryptDB

ANONYMITY

QUESTIONS

What is anonymity?
How can we achieve it?
How can we make it *usable*?

SYSTEMS

Tor, Mixnets
Dining cryptographers (DCNets)

ATTACKS

Fingerprinting attacks on Tor
Nation-state attackers

PAPERS

Tor	Users get routed
Mixnets	Fingerprinting

CENSORSHIP RESISTANCE

QUESTIONS Can we allow users to communicate despite powerful attackers trying to stop them? How does this relate to anonymity?

SYSTEMS Decoy routing (now “refraction routing”)
Alibi routing, DeTor

REPORTS “Enemies of the Internet”
by Reporters Without Borders

NETWORK SECURITY

QUESTIONS What can an attacker learn about two communicating hosts?

ATTACKS Malicious VPN apps (get on the path)

Off-path TCP attacks (side-channel attacks)

PAPERS Off-path TCP exploits

Measurement of VPN apps

BOTNETS

QUESTIONS

How do they operate?

What do they do?

How do we measure them?

IMPORTANCE

Botnets are a new, powerful force

All the more important due to IoT

PAPERS

Your botnet is my botnet

Understanding Mirai

DENIAL OF SERVICE (DOS) ATTACKS

QUESTIONS

How do we launch them?

How do we detect/measure them?

How do we *stop* them?

PAPERS

OptACK

Inferring DoS activity

IP Traceback

TVA