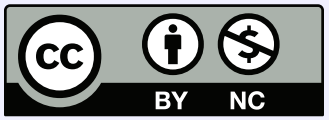
Protecting the Internet Against Large-Scale Passive Monitoring

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



The Internet is a Dangerous Neighborhood

- There are lots of hackers out there
- Eavesdroppers are watching every packet
- We have to use encryption all the time
- Or do we?
- Who is the enemy?

Types of Attack

Passive Attacker just listens

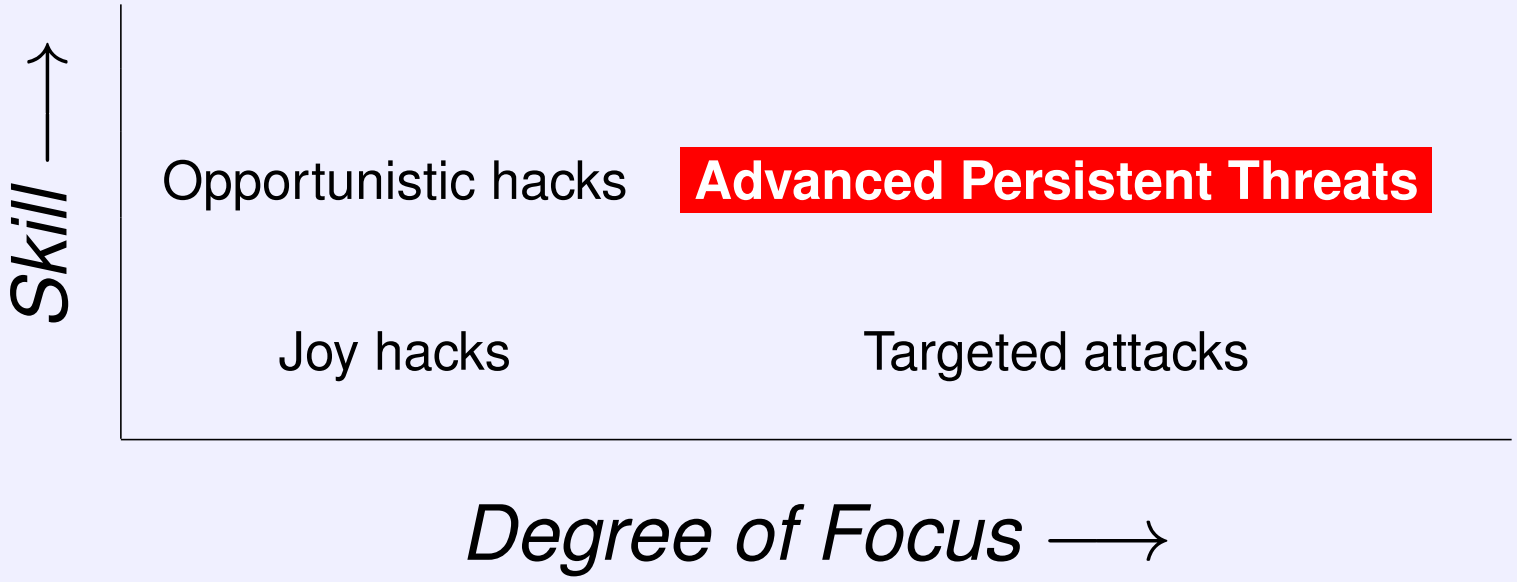
Active Attacker transmits messages; may be a “man in the middle” in a conversation

Targeted Only intercept traffic from a very few people or organizations

Large-Scale Trying to gather as much information as possible, from everyone, without trying to pick out particular targets

Our goal here is to defend against *large-scale, passive attackers*. These are typically government agencies of one sort or another.

The Threat Matrix



Who Can Eavesdrop on the Internet?

- WAN links are very hard to tap—not impossible, but very hard
- On-LAN monitoring is rather easy
- ☞ But if your LAN is secure, you don't have to worry about that
- ISPs can tap backbone and WAN links
- So can governments, with or without ISP cooperation
- Sophisticated attackers can inject false routes or DNS results—but those attacks are noticeable

Eavesdroppers

- Governments (especially major intelligence agencies) can do it
- Very sophisticated attackers—the ones in the top half of my chart—can do it
- Unskilled attackers can only do it if they penetrate your local LAN or a machine on it—but a penetrated machine is more likely to be used to attack other inside machines; eavesdropping by such malware is rare
- Physically being on a WiFi or Ethernet network requires physical proximity

☞ Conclusion: if you run a closed network—a house or an enterprise, or WiFi with WPA2—and need to worry about eavesdropping, your enemies will be very skilled. Simple defenses won't work.

Active Attacks Don't Scale Well

- The attacking machine must be on-path to every call—but it has bandwidth limits, too
- If it's remote, the increased latency will really hurt bandwidth
- Performance issues—lack of bandwidth, dropped packets, etc.—will degrade the real conversations
- Being on-path for lots of conversations is very difficult

☞ Conclusion: for large-scale attacks, the problem is *passive monitoring*

Large-Scale, Pervasive Monitoring: The Actual Threat Model

- Physically distant—an attacker can't be present everywhere
- Sophisticated attackers—easier attacks don't scale well
- Passive attacks only—simplifies our defenses
- (This is the upper-left quadrant of my chart)
- But—if you're targeted, *none* of these are true

Implications

- To prevent large-scale attacks, we don't have to worry about certain threats
- The goal is to make it *too expensive* to scan *your* traffic

☞ "Amateurs worry about algorithms; pros worry about economics"

- Cryptanalysis of modern algorithms is never free
- The goal is to conceal your traffic well enough to prevent you from being targeted

Example: Web Security

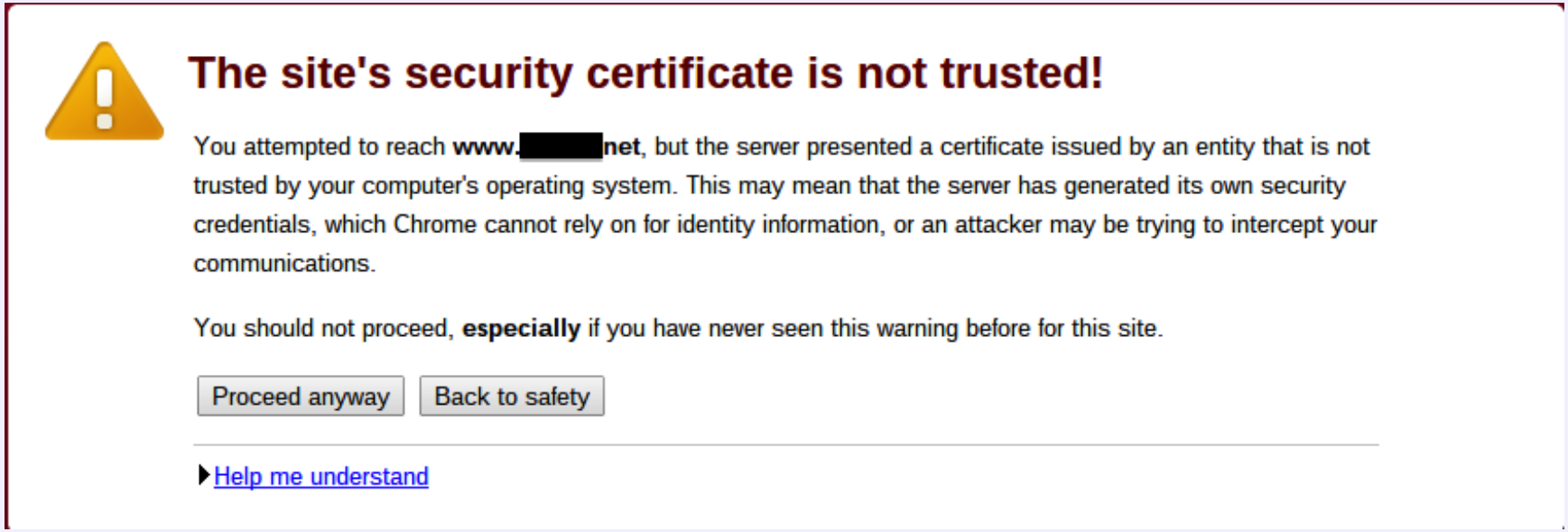
- Today: TLS with certificates

☞ Certificates are complicated

☞ Users care about security, but it's the web sites who buy them

☞ Users don't understand certificates or PKI

Certificate Warning Messages



This is completely incomprehensible!

Web Security with No Active Attacks

- A simple Diffie-Hellman or Elliptic Curve D-H exchange suffices (Diffie-Hellman key exchange sets up a secure but *unauthenticated* connection—you could be talking to anyone)
- We don't need certificates!
- Certificates are needed if and only if someone can impersonate the web site
- You need Diffie-Hellman even with certificates—if your private key is stolen, all conversations are readable unless you use Diffie-Hellman *and* certificate-based TLS
- But doing both is a lot more expensive, in CPU time and number of round trips
- We can't get rid of certificates—people still do banking on public networks—but frequently, they don't matter very much

Prevent Active Attacks is Hard!

There have been three major problems with SSL or TLS in just the last year:

goto fail Attack matters only if attacker sends fake certificate—irrelevant with passive attacks

Heartbleed Attack steals private key—which doesn't matter if D-H is used

POODLE Downgrade attack—again, requires active attack

None of these matter for *passive* attackers!

Hardening TCP

- Suppose that every TCP SYN packet contains a Diffie-Hellman exponential
- *All* TCP connections are then encrypted
- Encrypted traffic will no longer be unusual—today, the presence of encryption in an unusual context might be enough to get you targeted
- An active attacker can force a downgrade to unencrypted TCP—but that's an active attack

Can We Do It?

- The TCP options field will only allow for a small D-H modulus, about 128 bits—is that long enough?
- Discrete log is a “brittle” problem: with a lot of precomputation, any instance of it becomes very cheap. Is 128 bits good enough?

☞ We can’t negotiate different moduli

- What about TCP simultaneous open? (Extremely rare in practice)
- Is this too limited?

Tweaking TCP

- Instead of sending the exponential in the TCP option, negotiate the modulus
- "Steal" the first few hundred bytes of each TCP connection for the D-H exponential
- The code is a bit complicated but it should work
- We can protect all TCP connections against passive eavesdroppers, with *no* configuration necessary

Protecting Email

- A typical email message uses 3–6 TCP connections:
  - User to ISP server
  - (Probable: User IMAP connection to save copy of outbound message)
  - Sending ISP to receiving ISP
  - (Optional: internal ISP link to spam and/or virus scanner)
  - ISP server to user mailer
  - (Optional: IMAP copy to Trash folder)
- All of these are potentially vulnerable
- User connections to or from ISPs are frequently, but not always, encrypted
- ISP-to-ISP communications are rarely encrypted

Real End-to-End Email Security

- Conventional end-to-end email encryption requires certificates, but most people don't have them (or know what they are)
- Senders have no good ways to obtain recipients' certificates
- Encrypted email interferes with search

Obtaining Certificates

- All modern mailers support LDAP for consulting the local directory
- Instead of pointing at the organizational LDAP server, use an LDAP proxy
- It queries the LDAP server of each email recipient to retrieve the recipient's certificate
- Encryption can now happen

Lazy Certificate Generation

- Where do these certificates come from?
- If the user has one, it's easy
- If not—generate one when the query arrives, send it back
- Either send the private key to the recipient—or store it locally and decrypt email on receipt!

☞ Email is then encrypted from the sender to the recipient's ISP, and maybe all the way to the recipient's mailer

Problems with Encrypted Mail

- Not searchable
- Certificates expire
- Keys “age”—factoring algorithms improve over time
- Users forget the pass phrases to their old private keys

Re-Encrypting Email

- Download email from the IMAP server
- If it's already encrypted, decrypt it
- Store unencrypted email on the user's machine
- (Harden it, use full-disk encryption, etc.)
- ☞ Search now works, unchanged
- Re-encrypt with *today's* key, and rewrite to the server

Much More Secure!

- Incrementally deployable—each user can protect his/her own email from attacks on the IMAP server
- Works with encrypted or unencrypted email
- Works with old email, even email received before email encryption was invented
- Deals with aging algorithms: replace old RSA-512/DES-protected email with RSA-2048/AES versions
- Signatures are done on plaintext, before encryption; this doesn't disturb old signatures that you might need to show to a judge

IPsec

- Network-layer encryption
- Protects all traffic between two points
- Used for *virtual private networks* (VPNs)
- Two primary uses: connecting company offices to each other, and access to the corporate net for mobile devices

It's Hard to Configure

- IPsec is very hard—and needlessly hard—to configure
- From the **Racoon** man page (for IPsec's IKE key exchange):

```
send_cert (on | off);
   If you do not want to send a certificate, set this to
   off.  The default is on.
send_cr (on | off);]
   If you do not want to send a certificate request, set
   this to off.  The default is on.
```

- What does that mean? Why would you want to turn off certificates? Why would you want to turn off requests? What happens if you do? The documentation doesn't say!

Cryptographic Configuration

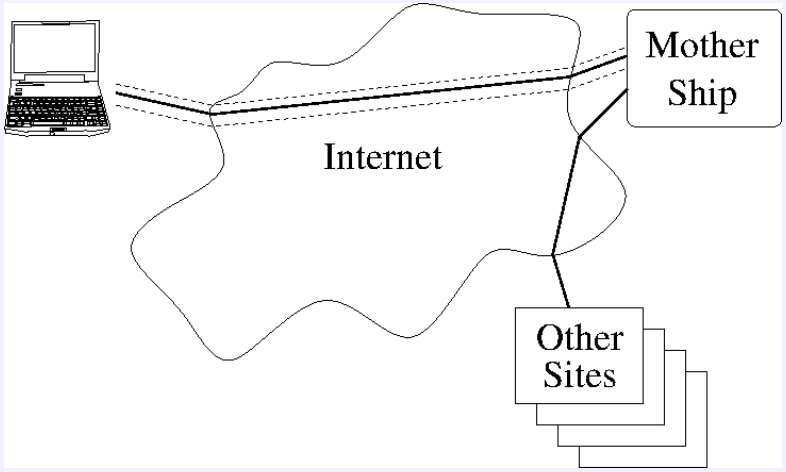
- Sysadmins have to pick cryptographic algorithms and key lengths
- Is AES-256 better than AES-128?
- When should you use elliptic curve instead of RSA?
- Do you need perfect forward secrecy?
- Most system administrators have no idea what the right answers are!

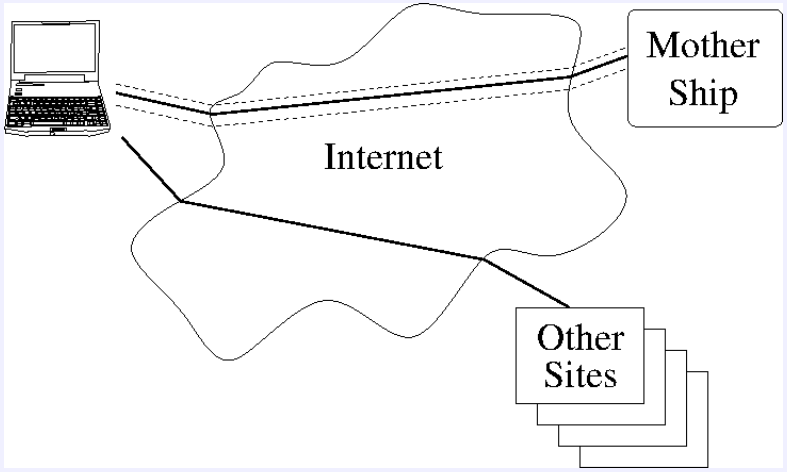
Why Expose Unnecessary Choices?

- The designers and implementors of cryptographic programs probably know the best answers
- (If they don't, they can find people who do.)
- The *protocol* needs algorithm agility, but *not* the implementation: modern algorithms do not collapse all at once. There's time to change the implementation.

IPsec Topologies

With IPsec, there is exactly one interesting policy choice: which topology is best?





The system administrator makes that one policy choice and describes the topology; everything else is “compiled” to IPsec configurations

Generalizing

- This principle is broadly true: do not force the users to make unnecessary choices
- Encryption (almost) never hurts; why not make it the default?
- It's true for email, it's true for TCP, it can be true almost everywhere
- There is rarely a noticeable performance penalty; today's computers are quite fast
- *Don't* ask people if it should be on; just do it!

Disadvantages?

- There are some elements—firewalls, network intrusion detection systems, a very few more—that do need to look at plaintext
- Often, this can be “outsourced” to the end-hosts: distributed firewalls, host IDSs, etc.
- Only a very few functions (such as switch snooping on IGMP announcements) are really hurt by ubiquitous encryption. Let’s redesign those protocols instead.

Defeating Large-Scale Passive Monitoring

- Even modest forms of encryption block large-scale passive attacks
- Our computers are powerful enough to make encryption the default
- At most, small protocol changes are needed
- Everything else can be done with a small amount of clever software