The Internet is Too Secure Already

Eric Rescorla

RTFM, Inc.

Overview of my argument

- We have lots of communications security tech.
  - RSA, AES, SHA-1, HMAC, IPsec, S/MIME, SSH, SSL
- But actual Internet communications are insecure
  - Why?
- We have the wrong threat model!
  - We worry about all known threats
  - Too good security is trumping deployment
  - Practical security isn’ t glamorous

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Better customer models

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Understand our customers

The Internet threat model

- Attacker has complete control of the network
  - Can modify, delete, insert, duplicate, etc.
  - "Hand packets to attacker to deliver"
- End-systems more or less inviolate
  - Not really true
  - ..but hard to do communications security without it
- Don' t get embarassed

Real attacks are less glamorous

- Remote penetration
  - Find simple programming bugs
    - Buffer overflows
    - Format strings…
  - Mostly a matter of effort…
- Malware
  - Viruses
  - Worms
- DDoS

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Understand our customers

Two wins, Three draws, One loss

- Before we can analyze we need data
- Two wins
  - SSL, SSH
- Three draws
  - IPsec, S/MIME, PKIX
- One loss
  - WEP

SSL/TLS Status

- Main protocol for Web security
  - And other kinds of channels
- Quite mature
  - SSLv2 released in 1994
    - Not very good
  - SSLv3 released in 1995
  - TLSv1 published in 1999
  - Now working on TLS 1.1
- TLS 1.1?
  - General cleanup + fix for Rogaway attack

SSL/TLS Deployment

- Very widely implemented
  - Nearly all browsers/servers have it
  - Lots of Open Source toolkits
- Usage
  - Web
    - Tens of billions of dollars in transactions
    - … but only about 1% of servers
      - And most of them have invalid certs
  - Non-web
    - Sporadic usage for SMTP, IM, etc.
      - Certificates almost always self-signed
- Most common use of TLS is for e-commerce
  - ..but credit card liability is limited
  - Free rider problem

Why has SSL succeeded?

- It's easy to use
  - Interface looks almost exactly like what it replaces
  - Just type "https" instead of "http"
- Can be deployed without much external help
  - Certificates are relatively easy to get
    - And only servers need them
  - Especially if you let mod_SSL make you a "Snake Oil" one
- There is a real incentive
  - Credit card sniffing was scary
  - ...and there was big money to be made

SSH Status

- Premier secure remote login protocol
  - Originally invented by Tatu Ylonen in 1994
  - Program was the spec
- Security status
  - Lots of holes in SSHv1
  - SSHv2 pretty good
- Standardization has really lagged
  - IETF standard version due out Real Soon Now
  - But the protocol is pretty mature now

SSH Deployment

- Near-universal on Unix
  - Available on Ciscos, etc.
  - Clients available for Windows, Mac, Java
- Arguably the most successful security protocol
  - Less total use than SSL
  - But completely dominates its market segment
    - Telnet and rsh have essentially vanished

Why has SSH succeeded?

- It’ s easy to use
  - Interface looks almost exactly like what it replaces
  - ..alias ‘ rsh’ to ‘ ssh’
- Can be deployed without any external help
  - Both parties to the transaction know each other
  - Leap of faith authentication
- There is a real incentive
  - Password sniffing was a real problem
  - Sysadmins can impose it on users
  - VPNs are a pain

SSH' s Leap of Faith

- Problem: client needs server' s public key
  - Don' t want to use certificates
- Solution:
  - Server gives client bare public key
    - MITM possible
    - Optional verification with fingerprints
  - Client caches server' s key
    - Detects changes
- This was not well received originally
  - But now it' s considered clever

IPsec Status

- Security for IP traffic
- Two main pieces
  - Packet formats
    - Authentication Header (AH)
    - Encapsulating Security Payload (ESP)
  - Key management: IKE (Internet Key Exchange)
- AH/ESP basically mature
  - Though getting tweaked
- IKE getting a total rewrite
  - This is **very** late.
    - Planned for 2001
    - Due Real Soon Now

IPsec Deployment

- Widely implemented
  - Built into Windows, Solaris, Cisco
  - Available for Linux (FreeS/WAN), FreeBSD (KAME)
- Only really used for VPNs
  - Using dedicated appliances
  - Manual configured
    - Shared static keys
    - Self-signed certs
  - Being replaced by SSL VPN!

S/MIME Status

- One of two primary e-mail encryption protocols
  - Designed by RSA
- S/MIME v2 stable and mature
- S/MIME v3 currently under development
- Minor tweaks only…
  - Replacing DH/DSS with RSA
    - Reversal of previous patent evasion
  - X.400 gatewaying
  - Symmetric key distribution

S/MIME Deployment

- In a number of major mail programs
  - Outlook, OE, Netscape
- Almost totally unused
  - PGP has more users
    - But not many
- Really hard to get certificates
  - Where can I get my own cert?
    - Verisign?
    - How hard is it?
      - It takes hours!
    - And what does it promise?
      - E-mail validity
      - I waited for that????
  - Other people' s certs are in hiding

Do people just not want secure e-mail?

- This is our third run at the wall!
  - At least…
    - PEM, MOSS are direct ancestors
  - Also PGP, DMS, X.400, OpenPGP
- Nobody wanted any of the others either
- But people say they want secure e-mail
  - And VCs believe it…
    - Voltage, PGP Inc., SIGABA, Tumbleweed
  - So what' s the story?

PKIX Status

- IETF Standard for certificates
- 8 years old
- Lots of output
  - 18 RFCs
  - 1.5 MB total
- And still plugging away
  - 28 I-Ds
  - 1.7 MB total
  - Plus, PKI Forum…
- Will we ever be done?

PKIX Deployment

- Lots of implementations
  - You get a CA for free with Windows Advanced Server!
  - But interoperability is a nightmare
  - Unless you stick to the common subset
- Internet deployment limited to SSL
  - And self-signed certs are common
- Enterprises bought PKI
  - But it made them miserable
  - …and they don’ t deploy it

The WEP Debacle

- "Security" for 802.11
- WEP is badly broken
- The big problem
  - The channel security misused RC4
    - Most common crypto error ever
  - Tools exist to break into any WEP network in minutes
- The small problem
  - Key management is simple shared key
    - Probably not the best idea
- These problems are being fixed
  - Still waiting for a final standard (TKIP, 802.11i)
  - Current deployed systems are broken

WEP Deployment Status

- In almost every 802.11 card and AP
- Not always turned on
  - 28 % of networks use it
    - And those networks are easily crackable
  - People seem to be scared by the publicity
- Still a lot more deployment than IPsec
- And a heck of a lot better than nothing

Common themes

- Use lags availability
  - Just having the stuff there isn’ t enough
- Certificates are really hard to get
  - Blocker for S/MIME, IPsec
  - Partial blocker for SSL
    - Wide use of weak certificates
- This stuff is too hard to use
  - See “ Why Johnny Can’ t Encrypt”
  - Do usage model first
    - Then get security right (SSL, SSH)

Some possible explanations

- Security is inherently hard to use
  - Possible but doesn’ t get us anywhere
- The customer is stupid
  - Probably true, but he’ s not getting any smarter
- We’ re delivering the wrong products
  - We’ ll sell no wine before its time
    - But we’ve been working on this stuff for > 10 years
  - We’ re using the wrong design criteria
    - So the end product is undesirable
    - **This is the only theory that gets us somewhere**

The wrong design criteria?

- This causes two kinds of problems
  - Intentionally building the wrong product
    - Because we think it' s the right one (IPsec)
  - Diverting resources due to feature misprioritization
    - Emphasis on security over usability (Name-based virtual hosts)
- Criteria cannot be derived from first principles
  - You have to know the customer

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Understand our customers

Where is the effort going?

- Inventing new mechanisms
  - Multicast
  - Stream authentication
  - New cipher modes
- Polishing existing protocols
  - Defenses against impractical attacks
  - New security features
  - Replacing old algorithms with new ones
    - OAEP, EC, CCM, XCBC, PSS
  - The occasional actual improvement

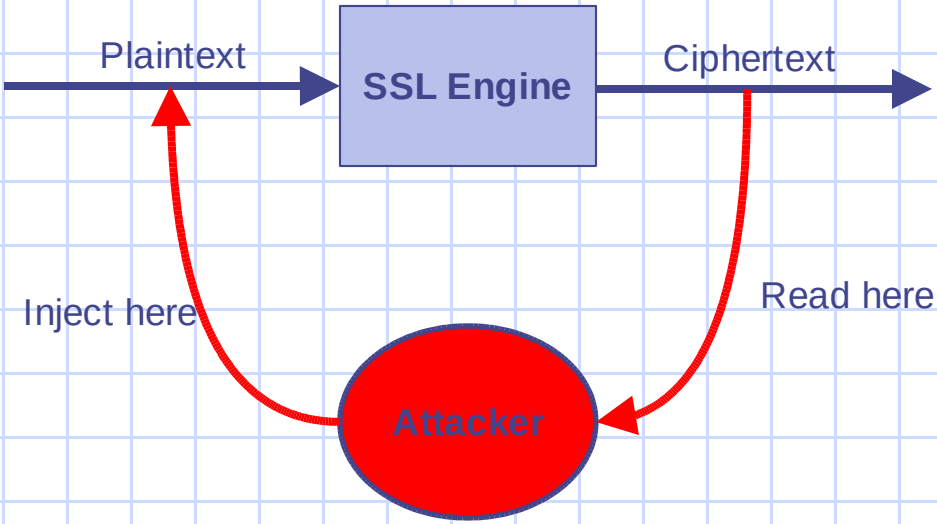
Name-based virtual hosting

- HTTP virtual servers
  - Multiple web servers on a single physical server
  - Disambiguated by the Host: header
- But don’ t work with HTTPS
  - Need to know virtual host to choose certificate
  - But SSL handshake happens first
    - So you don’t see the Host: header till too late
  - With SSL you need 1 IP address per virtual host
- Fix: put the name in the SSL handshake
  - Done in Domain name extension
  - But held hostage to..
    - Packet size, external certs, OCSP…

Current work on SSL/TLS: Attacks

- Kocher/Boneh/Brumley timing attack
  - Extract a private key
  - But how practical is it?
    - several million trials on an intranet
    - Billions on a WAN?
  - OpenSSL finally fixes it…
- Vaudenay CBC attack
  - Extract passwords from automated clients
- Rogaway CBC attack
  - Verify a guess of a single cipher block
- Bad Version Oracles
  - Recover a single session key
    - Extension to Bleichenbacher's attack
  - Requires a million trials

Rogaway CBC Attack



- Attacker can verify guess of ciphertext
  - By injecting a chosen plaintext
  - And observing
- This only works well when SSL is used in a proxy

Current work on SSL/TLS: Responses to Attacks

- Wide publicity
  - Most of these attacks lead to papers
  - Vaudenay, Kocher/Boneh/Brumley attacks got coverage
- Immediate fixes issued
  - To OpenSSL
  - New version of TLS
- No known actual attacks in wild
  - No known available tools
  - Contrast with OpenSSL buffer overflows
    - Slapper released within 2 months

Current work on IPsec

- AH/ESP are basically unchanged
- IKE being totally redone (IKEv2)
  - This has taken 2 years!
- What issues are holding us up?
  - Cipher suites vs. a la carte
  - Identity protection?
  - 6 messages or 4
  - Provable exchange
- Did I mention it still doesn’ t work?
  - Certificates and fragmentation
- I am not making this up!

Why do IKEv2 anyway?

- Nobody wants IKEv1
  - Complaints that it' s too hard to implement
    - Vague specification
    - Extremely complex protocol
- ..but there are lots of interoperable implementations
  - VPNC lists >10 conformant implementations
- The real reason?
  - We' re flailing
    - Nobody uses IKE
  - ..so we have to try *something*

What’ s the story with S/MIME?

- The protocol is in good shape
- Everyone has it, noone uses it
- The problem is certificates
  - Required
  - But noone has them

PKIX, the standard that won’ t die

- Hideously complex
  - RFC 3280 is 129 pages long
  - Lawyers are involved!
- Noone knows what anything means
  - DNs
    - Comparison
    - Structure
  - keyUsage (nonRepudiation)
  - Constraints
  - Policies
- And they don’ t implement it anyway
  - CRL checking
  - Constraints again
- All I wanted was to authenticate who I was talking to!

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Understand our customers

Three examples of threat model mismatch

- Excessive concern with active attacks
  - The easiest attacks are passive
  - Leads to requirement for certificates
- Taking cryptanalytic attacks too seriously
  - Leads to protocol churn
    - Not bad in itself, but very distracting
- Forgetting about other threats
  - User stupidity
  - Software holes

Why isn’ t there any tooling to steal private keys?

- Kocher’ s timing attack is years old
  - But no tools are available
  - We know it’ s possible now…
    - But still no tooling
  - Maybe it’ s still too hard?
- The OpenSSL exploits didn’ t steal private keys
  - This would have been incredibly easy
    - The keys are generally just in memory
- Other exploits don’ t seem to either
  - When the Web server is compromised it’ s easy
- Maybe people don’ t want them…

Maybe private keys aren’ t so important?

- Using a stolen key is harder than it looks
  - Pretty much requires being on the same network as victim
- People’ s information isn’ t that interesting
  - Credit card numbers are easy to get
    - Buy on the black market
    - Break into e-commerce servers
  - SSH keys are only useful for breakins
    - But once you’ ve already broken in…

The worst case happens.. And noone notices!

- What happens if a CA is compromised?
  - An attacker can impersonate everyone
    - Pretty bad, huh?
- IE cert verification was totally broken until 2002
  - Basic constraints verification broken
    - This means that anyone can forge certificates!
  - This is worse than a CA compromise
    - Since it can't be fixed with CRLs
  - Lots of people still have broken versions
    - Because they haven't upgraded
- And yet no rash of attacks

User Stupidity (I): SSL Certificates

Clicking "Yes" may not be your best plan here!

User Stupidity (II): Executable email

- Windows allows executable email
  - VBScript
  - Javascript
  - Actual Windows binaries
- Users are asked before .EXEs are run
  - And they often say yes
  - Worms often spread this way
- How can secure e-mail work in this environment?

Bugs in software

"All software has bugs. Security software has security-relevant bugs"

-- Steve Bellovin

- Holes found in most COMSEC implementations
  - Buffer overflows
    - OpenSSL
    - OpenSSH
    - IE
    - IIS
    - …
  - Failure to correctly perform protocols
    - IE
    - GPG
    - …

What' s an appropriate threat model?

- Worry a lot about passive attack
  - Else why bother at all?
- Worry about active attack
  - But not if it means making things undeployable
  - Lesson of SSH
    - Leap of faith
- Don' t worry at all about being embarrassed
  - Unless you did something really stupid

Structure of this talk

- Overview of the current situation
  - Internet threat model
  - Real protocol deployment
- Where is the effort going and why?
- How could we make things better?
  - Appropriate threat models
  - Understand our customers

Customers lie

- But not all the time
- Our job is to know what they want
  - and to try to give it to them
  - or they won’ t take it...
- What they want may not be what we think they should want

“ Security is really important”

this means...

“ Security is really important”

this means...

“ The appearance of security is really important”

- These are not the same thing
- He wants to know what to tell his boss

“ Security is more important than features”

this means...

“ Security is more important than features”

this means...

“ I want my dancing pigs”

- In the battle between features and security features always win

  Active content, firewall bypassing, Windows...

- Don’ t torture your users

“ Make security easy to install”

this means...

“ Make security easy to install”

this means...

“ It better just drop in and work”

- Need I say more?

An Agenda: Evidence-based Security

- General problem: what security measures make a difference
  - What threats are most serious?
  - Which ones can we fix
    - And at what cost?
  - What will users deploy?
- These questions can’ t be answered a priori
  - It requires unglamorous research

What threats are most serious?

- This is probably where we have the most data
  - Market research
    - … but it's spotty
    - Companies tend to hide this information
  - Not too much academic research
- My impressions
  - Cryptanalytic attacks are really rare
  - Protocol flaws are rare
  - Attacks on programming flaws are common
  - DDoS is common

Which threats are easy to fix?

- Some gut reactions
  - Our protocols are about as secure as they' re going to get
    - But we can make them more deployable
  - But we can fix our code problems
    - Stop using C
    - Sandboxing, compiled in protection
      - Janus, Systrace, *guard
    - Code checking tools
      - Metacompilation, RATS, Lint…
  - We don' t know how to really fix DDoS
- This is going to take some measurement
  - User/Programmer experience
  - System performance

What will the customers buy?

- Rule of thumb: if we' ve spent a lot of time on it and noone wants it, something is wrong
  - IPsec, X.509, secure e-mail
  - Painting it a different color probably won' t help
- Compliance is key
  - Side effects
  - Perceived cost/benefit ratio
    - Users may not make the choices we would

Questions we need to answer

- What’ s the total cost of exposure of various kinds of threats?
- How much are people willing to pay for various security features?
- Why can't users use cryptographic protocols?
- What percentage of security protocol features see implementation?
- What sorts of implementation errors are most serious?
- What programming practices would minimize them?
- What's the cost of upgrades?
- What's the cost of obtaining information about vulnerabilities?
- What sort of incentives would cause users to keep up to date?