Surveillance, Censorship, and Countermeasures

Professor Ristenpart

http://www.cs.wisc.edu/~rist/

rist at cs dot wisc dot edu
AT&T Wiretap case

- Mark Klein discloses potential wiretapping activities by NSA at San Francisco AT&T office
- Fiber optic splitter on major trunk line for Internet communications
  - Electronic voice and data communications copied to “secret room”
  - Narus STA 6400 device
Large amounts of Internet traffic cross relatively few key points
Interception technology

• From Narus’ website (http://narus.com/index.php/product/narusinsight-intercept):
  – “Target by phone number, URI, email account, user name, keyword, protocol, application and more”, “Service- and network agnostic”, “IPV 6 ready”
  – Collects at wire speeds beyond 10 Gbps
Types of packet inspection

Internet service providers need only look at IP headers.

Deep packet inspection (DPI) analyzes application headers and data.
Is dragnet surveillance technologically feasible?

• CAIDA has lots of great resources for researchers about traffic levels

• From their SanJoseA tier-1 backbone tap:

```
<table>
<thead>
<tr>
<th>Application</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>51.20M</td>
<td>2.20G</td>
<td>11.01G</td>
</tr>
<tr>
<td>UNKNOWN_UDP</td>
<td>4.08M</td>
<td>168.79M</td>
<td>711.57M</td>
</tr>
<tr>
<td>UNKNOWN_TCP</td>
<td>3.62M</td>
<td>136.02M</td>
<td>660.50M</td>
</tr>
<tr>
<td>HTTPS</td>
<td>3.96M</td>
<td>125.80M</td>
<td>543.15M</td>
</tr>
<tr>
<td>RTMP</td>
<td>2.00M</td>
<td>78.09M</td>
<td>314.79M</td>
</tr>
<tr>
<td>SMTP</td>
<td>289.75k</td>
<td>14.76M</td>
<td>55.82M</td>
</tr>
<tr>
<td>QUAKE</td>
<td>300.58k</td>
<td>8.31M</td>
<td>36.02M</td>
</tr>
<tr>
<td>SQUID</td>
<td>42.88k</td>
<td>7.25M</td>
<td>37.58M</td>
</tr>
<tr>
<td>IPSEC</td>
<td>213.15k</td>
<td>7.09M</td>
<td>23.97M</td>
</tr>
<tr>
<td>SSH</td>
<td>248.25k</td>
<td>6.73M</td>
<td>28.40M</td>
</tr>
<tr>
<td>WOW</td>
<td>72.88k</td>
<td>6.12M</td>
<td>34.40M</td>
</tr>
<tr>
<td>ABACAST</td>
<td>285.74k</td>
<td>3.43M</td>
<td>14.98M</td>
</tr>
<tr>
<td>NOPORTS_UDP</td>
<td>64.46k</td>
<td>2.04M</td>
<td>14.83M</td>
</tr>
<tr>
<td>other</td>
<td>1.23M</td>
<td>40.23M</td>
<td>161.56M</td>
</tr>
</tbody>
</table>
```

http://www.caida.org/data/realtime/passive/?monitor=equinix-sanjose-dirA
Key Features

Precision Targeting at Broadband Speeds

- Broad range of target types from Layer 2 through Layer 7, including ATM/MPLS/VPN support
- Target by phone number, URI, email account, user name, keyword, protocol, application and more
- Service- and network agnostic
- IPV 6 ready

Capture and Delivery

- Passive model collects from the line at wire speeds beyond 10 Gbps with support for asymmetric networks
- Efficient encoding of full packets and associated metadata for economical backhaul
- Flexible delivery for remote monitoring, retention or forwarding to alternate agencies

Reconstruction and Rendering

- Reconstruction and playback of captured traffic in near real time
- Integrated rendering of voice, video, email, Web mail, chat, and more
- Access to extensive metadata for all traffic types
Lawful intercept

• CALEA
  – Communications Assistance for Law Enforcement Act (1995)
• FISA
  – Foreign Intelligence Surveillance Act (1978)
  – Demark boundaries of domestic vs. foreign intelligence gathering
  – Foreign Intelligence Surveillance Court (FISC) provides warrant oversight (good example of regulatory capture)
  – Executive order by President Bush suspend need for NSA to get warrants from FISC
• Almost all national governments mandate some kind of lawful intercept capabilities
Lots of companies

• Narus (originally Israeli company), now owned by Boeing
  – Partnered with Egyptian company Giza Systems
• Pen-Link (http://www.penlink.com/)
• Nokia, Nokia Siemens
• Cisco
• …
NarusInsight™ Selected To Save Pakistan's Telecommunications Networks Millions Of Dollars Per Year

NarusInsight™ Selected to Save Pakistan’s Telecommunications Networks Millions of Dollars Per Year

Narus System Chosen to Detect Rogue VoIP Traffic

MOUNTAIN VIEW, Calif.—September 21, 2007—Narus, Inc., the leader in carrier-class security for the world’s largest IP networks, today announced that the company has teamed up with Inbox Business Technologies Pvt. Ltd, a leading total IT solution provider in Pakistan, to keep Pakistan’s telecommunication networks clear of illegal, rogue and malicious IP traffic. NarusInsight was chosen by the Pakistan Telecommunication Authority (PTA) (the government administration responsible for regulating the establishment, operation and maintenance of telecommunication systems, and the provision of telecom services) to detect rogue VoIP traffic flowing through the telecommunications network in Pakistan.

Preventing intercept

• End-to-end encryption (TLS, SSH)

• What does this protect? What does it leak?

• What can go wrong?
End-run around HTTPS

- HTTPS terminated at edge of Google networks
- Internal data center-to-data center communications on privately leased lines
  - No encryption up until summer 2013
Sabotaging TLS

• NIST’s Dual EC pseudorandom number generator (PRNG) apparently backdoored
  – Mandated public parameters are public key
  – There exists a secret key, the trapdoor

• One output of PRNG + trapdoor reveals next state of PRNG, and prediction of future outputs
TLS handshake for RSA transport

Bank customer → Bank

1. Pick random Nc
2. Check CERT using CA public verification key
3. Pick random PMS
4. C ← E(pk, PMS)
5. Pick random Ns
6. CERT = (pk of bank, signature over it)
7. ClientHello, MaxVer, Nc, Ciphers/CompMethods
8. ServerHello, Ver, Ns, SessionID, Cipher/CompMethod
9. PMS ← D(sk, C)
10. Say client is using Dual EC for randomness generation

What is vulnerable?

- RSA BSAFE library: 2.4 seconds to recover PMS
- Windows: 60 minutes
- OpenSSL: never (bug in code!)

See http://dualec.org/
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
    uint8_t *signature, UInt16 signatureLen)
{
    OSStatus err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
Preventing intercept

• End-to-end encryption (TLS, SSH)

• What does this protect? What does it leak?
• What can go wrong?
Hiding “metadata” such as connectivity is even harder

- IP addresses are required to route communication, yet not encrypted by normal end-to-end encryption
  - 1.2.3.4 talked to 5.6.7.8 over HTTPs
- How can we hide connectivity information?
Anonymization systems

• Single-hop proxy services

• JonDonym, anonymous remailers (MixMaster, MixMinion), many more...

Thursday, April 26, 2012

FBI seizes server used to anonymize e-mail

Jeffrey Brown 1 comment
Tor (The Onion Router)

IP: 1.2.3.4

IP: 5.6.7.8

Other major backbone

AT&T network

Interception gear

Tor Node 7.8.9.1

Tor Node 9.1.1.2

Tor Node 8.9.1.1
Onion routing: the basic idea

Tor implements more complex version of this basic idea
What does adversary see?

Tor obfuscates who talked to who, need end-to-end encryption (e.g., HTTPS) to protect payload
Beyond Silk Road 2.0, over 400 'dark web' Tor sites seized by FBI

Summary: Hundreds of darknet websites have been identified and taken down -- and the Tor Project isn't sure how.
Tor hidden services

Other major backbone

IP: 1.2.3.4

Tor Node: 7.8.9.1

AT&T network

Interception gear

IP: 5.6.7.8

Tor Node: 8.9.1.1

Tor Node: 9.1.1.2
Surveillance via third-party

• “Thus, some Supreme Court cases have held that you have no reasonable expectation of privacy in information you have "knowingly exposed" to a third party — for example, bank records or records of telephone numbers you have dialed — even if you intended for that third party to keep the information secret. In other words, by engaging in transactions with your bank or communicating phone numbers to your phone company for the purpose of connecting a call, you’ve "assumed the risk" that they will share that information with the government.”

From the EFF website
https://ssd.eff.org/your-computer/govt/privacy
Third-party legal issues

• Under Electronic Communications Privacy Act (ECPA) government has access via subpoena to:
  – Name, address
  – Length of time using service
  – Phone records (who you called, when, how long)
  – Internet records (what/when/how long services you used, your assigned IP address)
  – Info on how you pay your bill
Example: AT&T Hawkeye database

• All phone calls made over AT&T networks since approximately 2001
  – Originating phone number
  – Terminating phone number
  – Time and length of each call
Example: Google data requests

<table>
<thead>
<tr>
<th>Country</th>
<th>User Data Requests</th>
<th>Percentage of requests where some data produced</th>
<th>Users/Accounts Specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>10,574</td>
<td>83%</td>
<td>18,254</td>
</tr>
<tr>
<td>France</td>
<td>2,750</td>
<td>51%</td>
<td>3,378</td>
</tr>
<tr>
<td>Germany</td>
<td>2,660</td>
<td>40%</td>
<td>3,255</td>
</tr>
<tr>
<td>India</td>
<td>2,513</td>
<td>66%</td>
<td>4,401</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>1,397</td>
<td>69%</td>
<td>3,142</td>
</tr>
<tr>
<td>Brazil</td>
<td>1,085</td>
<td>49%</td>
<td>1,471</td>
</tr>
<tr>
<td>Italy</td>
<td>896</td>
<td>42%</td>
<td>1,084</td>
</tr>
<tr>
<td>Australia</td>
<td>780</td>
<td>70%</td>
<td>944</td>
</tr>
<tr>
<td>Singapore</td>
<td>755</td>
<td>68%</td>
<td>847</td>
</tr>
<tr>
<td>Spain</td>
<td>545</td>
<td>53%</td>
<td>761</td>
</tr>
<tr>
<td>Poland</td>
<td>502</td>
<td>23%</td>
<td>740</td>
</tr>
<tr>
<td>Taiwan</td>
<td>439</td>
<td>61%</td>
<td>580</td>
</tr>
</tbody>
</table>

July to December 2013

From http://www.google.com/transparencyreport/governmentrequests/userdata/
Should we prevent? Can we?

- One can encrypt data that is stored, but no current way to protect data that needs to be used
- Companies are increasingly worried about perception of government surveillance
- Policy?
- Legal protections?
Censorship
Censorship via Internet filtering

• Golden Shield Project most famous example
• But many other nations perform filtering as well including
  • Iran, Syria, Pakistan (YouTube anecdote)
  • Turkey (twitter ban recently)
  • Singapore, Australia (proposed legislation)
• Other countries?
Big business

• Reports of products being used in Syria
  – Blue Coat (http://www.bluecoat.com/)
  – NetApp (http://www.netapp.com/)

• Iran, Saudi Arabia
  – Secure Computing’s SmartFilter software
  – Secure Computing recently bought by McAffee

• Embargos prevent selling directly by USA companies, but resellers end up doing so
Filtering

- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
- Protocol filtering (detect Tor protocol)
Circumvention of filtering

- IP filtering
  - Proxies
- DNS filtering / redirection
  - DNS proxy
- URL filtering or Packet filtering
  - Encryption / Tunneling / obfuscation
- Protocol filtering
  - Obfuscation techniques
Golden Shield Project (Great Firewall of China)

- IP filtering
- DNS filtering / redirection
- URL filtering
- Packet filtering (search keywords in TCP packets)
  - Send TCP FIN both ways
- Protocol filtering (Tor is shut down)
Great Firewall targeting of Tor (circa 2011 and before)

• Enumerate Tor relays and filter them

Relay is publicly listed Tor node

Bridge is Tor node not publicly listed

The Tor Project - https://metrics.torproject.org/
Number of directory requests to directory mirror trusted

https://torproject.org

China
Chinese Tor users via bridges

Tor project -- www.torproject.org
Great Firewall targeting of Tor (circa 2011-2012)

TLS connections with particular ciphersuites flagged

From [Winter, Lindskog 2012]

https://gist.github.com/da3c7a9af01d74cd7de7
Tor uses TLS for point-to-point communications, including first hop.

Tor clients used relatively non-standard Ciphers.
Great Firewall targeting of Tor (circa 2011-2012)

TLS connections with particular ciphersuites flagged

Attempt to connect to dest IP by Tor client (source IP may be spoofed)

If server speaks Tor, then IP added to GFW black list

From [Winter, Lindskog 2012]

https://gist.github.com/da3c7a9af01d74cd7de7
Islamic Republic of Iran

• Every ISP must run “content-control software”
  – SmartFilter (up until 2009)
  – Nokia Siemens DPI systems

• According to wikipedia Facebook, Myspace, Twitter, Youtube, Rapidshare, Wordpress, BBC, CNN, all have been filtered

• Occassional widespread filtering of Tor, TLS, other encrypted protocols
Iran DPI to shut down Tor

- Tor makes first hop look like TLS/HTTPS connection
TLS Handshake

ClientHello, MaxVer, Nc, Ciphers/CompMethods

ServerHello, Ver, Ns, SessionID, Cipher/CompMethod

CERT = (pk of bank, signature over it)

C

ChangeCipherSpec,
{ Finished, PRF(MS, “Client finished” || H(transcript)) }

ChangeCipherSpec,
{ Finished, PRF(MS, “Server finished” || H(transcript’)) }

MS <- PRF(PMS, “master secret” || Nc || Ns )

Bracket notation means contents encrypted
Iran DPI to shut down Tor

• Tor makes first hop look like TLS/HTTPS connection

• Use DPI to filter Tor connections:
  – Tor certificates have short expiration date
  – Most websites have long expiration date
  – Shut down those connections with short expiration dates

• Tor fixed via longer expiration dates

• Later in 2012: blocking/degrading all TLS connections
Directly connecting users from Egypt

The Tor Project - https://metrics.torproject.org/
From BlueCoat:

- Our awareness of the presence of these ProxySG appliances in Syria came from reviewing online posts made by so-called “hacktivists” that contained logs of internet usage which appear to be generated by ProxySG appliances. We believe that these logs were obtained by hacking into one or more unsecured third-party servers where the log files were exported and stored. We have verified that the logs likely were generated by ProxySG appliances and that these appliances have IP addresses generally assigned to Syria. We do not know who is using the appliances or exactly how they are being used. We currently are conducting an internal review and also are working directly with appropriate government agencies to provide information on this unlawful diversion.
Directly connecting users from the Syrian Arab Republic

The Tor Project - https://metrics.torproject.org/
“Twitter, mwitter!”

Directly connecting users from Turkey

The Tor Project - https://metrics.torproject.org/
Censorship summary
Protocol identification via deep-packet inspection (DPI)

DPI users want to identify protocol X

\( X = \text{TLS or Tor} \) then throttle connection

\( X = \text{HTTP} \) then leave it alone

\( X = \text{???} \) then throttle traffic

Check packet contents against *regular expressions*

\(^/\text{~\text{x16\text{x03[\text{x00\text{x01\text{x02}]..\text{x02...\text{x03[\text{x00\text{x01\text{x02]}|...? \ .*/}~}~}

Free translation: Does packet include “I’m TLS 1.1” ?
Scenario:
DPI system only allows HTTP traffic unfettered

Tor client → DPI system → Tor proxy

Steganography (e.g., Stegotorus): embed bits into HTTP messages
- Too slow for practical use (56k modem anyone?)

Obsfproxy (built into Tor): encrypt all bits sent over network (no plaintext bits)
- Really fast
- But DPI will flag traffic as ???

Want way to force DPI to classify traffic incorrectly as HTTP
So-called “misclassification attacks” against DPI
Surveying modern DPI systems

<table>
<thead>
<tr>
<th>System</th>
<th>Look at ports?</th>
<th>TCP stream reassembly</th>
<th>Uses regex’s</th>
<th>Use’s C/C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppID</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>L7-filter</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Yaf</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Bro</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>nProbe</td>
<td>No</td>
<td>Yes</td>
<td>Not explicitly</td>
<td>Yes</td>
</tr>
<tr>
<td>Proprietary*</td>
<td>Yes</td>
<td>Yes</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

* Hint: it’s a serious product (~$10k) and similar ones seem to be used in Iran.

Can we build encryption schemes that fool regex-based systems?
Attacking DPI

Cryptographic secret key $K$

$(\text{Conventional}) \quad \text{Encryption}$

$M \rightarrow K \rightarrow C$

$C \text{ looks like random junk. Won't look like HTTP}$

[Dyer, Coull, R., Shrimpton – CCS 2013]
Attacking DPI

Cryptographic secret key $K$

Regular expression $R$ specifying desired *ciphertext format*

$K$, $R$

Format-Transforming Encryption

$M$ $\rightarrow$ $C$

$C$ is guaranteed to match against $R$

[Dyer, Coull, R., Shrimpton – CCS 2013]
Realizing regex-based FTE

How should we realize regex-based FTE?

We want: Cryptographic protection for the plaintext Ciphertexts in L(R)
Realizing regex-based FTE

How should we realize regex-based FTE?

We want: Cryptographic protection for the plaintext Ciphertexts in L(R)
Let $L(R)$ be lexicographically ordered $x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|-1}$

Given a **DFA** (deterministic finite automaton) for $L(R)$, there are **efficient** algorithms
Let $L(R)$ be lexicographically ordered
\[ x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|-1} \]

Given a \textbf{DFA} (deterministic finite automaton) for $L(R)$, there are \textbf{efficient} algorithms

\[
\text{rank: } L(R) \longrightarrow \{0, 1, \ldots, |L(R)|-1\}
\]
Ranking a Regular Language

Let \( L(R) \) be lexicographically ordered
\( x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|-1} \)

Given a DFA (deterministic finite automaton) for \( L(R) \), there are efficient algorithms

\[
\begin{align*}
\text{rank}: & \quad L(R) \rightarrow \{0, 1, \ldots, |L(R)|-1\} \\
\text{unrank}: & \quad \{0, 1, \ldots, |L(R)|-1\} \rightarrow L(R)
\end{align*}
\]

such that \( \text{rank}(\text{unrank}(i)) = i \)
and \( \text{unrank}(\text{rank}(x_i)) = x_i \)

With precomputed tables, rank, unrank are \( O(n) \)

[Goldberg, Sipser ’85]
[Bellare et al. ’09]
Realizing regex-based FTE

Intermediate ciphertext, interpreted as an integer $i$...

Regex $R$\n
NFA $M$\n
DFA $M'$

Exponential blow-up in worst case. Regexes we needed avoid this.

FTE using NFAs directly

[Luchaup, Dyer, Jha, R., Shrimpton – In submission 2014]
We built a complete FTE record layer and proxy system

Client of protocol X

FTE client

FTE(K,R₁, M₁)

FTE(K,R₂, M₂)

FTE server

Server of protocol X

You are protocol Y!

Want to trick DPI into thinking we’re protocol Y != X
Where do we get R₁ and R₂?

(1) Get from DPI themselves
(2) Easy to manually craft
(3) Learn from traffic samples

We built regexes for variety of “cover” protocols:
Y = HTTP, SSH, SMB, SIP, RTSP
Evaluating FTE

Client of protocol X

Server of protocol X

FTE client $\rightarrow$ FTE($K,R_1,M_1$) $\rightarrow$ FTE($K,R_2,M_2$) $\rightarrow$ FTE server

Tests with gets on Alexa Top 50 sites ($X = \text{mix of HTTPS/HTTP}$)

$R_1 \, R_2$ set to HTTP, SSH, SMB, and more. When do we trick DPI?

<table>
<thead>
<tr>
<th>System</th>
<th>DPI-derived regex’s</th>
<th>Manual regex’s</th>
<th>Learned regex’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppID</td>
<td>Always</td>
<td>Always</td>
<td>Always</td>
</tr>
<tr>
<td>L7-filter</td>
<td>Always</td>
<td>Always</td>
<td>Always</td>
</tr>
<tr>
<td>Yaf</td>
<td>Always</td>
<td>Always</td>
<td>Always</td>
</tr>
<tr>
<td>Bro</td>
<td>Sometimes</td>
<td>Always</td>
<td>Always</td>
</tr>
<tr>
<td>nProbe</td>
<td>Never</td>
<td>Always</td>
<td>Almost always</td>
</tr>
<tr>
<td>Proprietary</td>
<td>Always</td>
<td>Always</td>
<td>Always</td>
</tr>
</tbody>
</table>
Web-browsing performance

**Punchline:** FTE or SSH tunnel result in the same user web-browsing experience
A field test...

Without FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…

With FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…

Used FTE to download Tor bundle:

Tor without FTE: “active blacklisting” attack on proxy
Tor through FTE: no problems

Ran various tests every 5 minutes for one month, no sign of detection in logs. (We shut it down after that.)
FTE is open source, runs on multiple platforms/OS, and fully integrated into Tor.

http://fteproxy.org