Reflections on trusting the same-origin policy

... and other web+network trust issues
me

The North American Network Operators’ Group

General Information

- What Is NANOG?
- Charter
- NANOG Endorsed by FARNET Membership, 26 October 1994
- Other Regional Networking Organizations

Meetings

- Next Meeting: Tentatively scheduled for October 27-28 in Phoenix!
- Past Meetings and Presentations
- Tips for Hosting a NANOG Meeting
- Search Engine for the NANOG Notes
  From Stan Barber, Academ Consulting Services

Mailing List

- Charter/AUP
- How to Join the NANOG Mailing List
- Searchable NANOG Mailing List Archive from CCTec
- NANOG Mailing List Archives from Merit

- Andre Gironda
- 15 years Unix+Internet experience
- First ran `www` in summer of 1993
- Cisco certified, NANOG attendee since ’97
  - That means I’m a network guy more than a systems guy, programming guy, or security guy
- Phoenix native
Past work

Vulnerability Research at lockdatasystems since 1997

- CatOS version 2.1
- IPv4 Vulnerability

Several Cisco zero-day denial-of-service (IOS, CatOS)
I’m bringing network back

- Routers are sexy
- Webapp security doesn’t make network security obsolete or any less interesting / important
- There are plenty of innovations left in both camps
- There is plenty of cross-over between the two
BGP

- Routing protocol
- 1993 CIDR
- Attributes
  - Well-known or optional
    - Well-known: Origin, AS-Path, Next-hop, local-preference
  - Transitive or non-transitive (local)
    - Optional and transitive: Community
    - Optional and non-transitive: MED

Influence your Neighbor’s Neighbor
- Can be used to give clues to peers and beyond
- Intermediate ASs must allow color to transit
MPLS

- Enno Rey: LayerOne 2006 – MPLS Security
- Injection of label traffic from the Internet
Wireless

Airport ad-hoc viruses + Karma = World MITM

– First New York, then San Francisco, New Orleans, Rio de Janeiro, Rome, Kinshasa, Karachi, Bangkok, and Beijing (from the movie, 12 Monkeys)
This old vulnerability: CLID

Caller ID trust

- Spoofing CLID
  - Lax VoIP providers
  - Security screens (a misnomer?)
- Spoofing ANI
- Beige-boxing
Network Vulns: borken stuff

- Original SYN attack still works.. (juno-z)
- Mitnick hack: rlogin
  - Hosts.equiv, .rhosts
  - AIX rlogin –froot
  - Solaris 10/11: 0day was the case that they gave me
    - `telnet –l “-froot” 10.0.0.1`
- Smurf attack, DDoS
  - ICMP reflection
  - TCP amplification

[Full-disclosure] "0day was the case that they gave me"
Advanced Network Vulns: trust

- **BGP nastiness**
  - Jack moves, AS loops, Stealing traffic

- **Proxies**
  - Tor (we’ll get to this a little later…)
  - Firewall hole punching (Skype, chownat)

- **MITM**

  Arp poisoning, DNS hijacking, cache poisoning, Wireless, STP takeover, HSRP/VRRP takeover, MPLS VPN / VPLS label modification / injection, VLAN / VPN hopping, CAM table overflow, Slipping windows

---

**Abstract: Peering Dragnet: Examining BGP Routes Received from Peers**

Tom Scholl & Aman Shaikh, AT&T Labs
Nathan Patrick, Sonic
Richard Steunenberg, Layer

In the hot-potato settlement-free peering world of today, there is an expectation that all peers play & advertise routes equally. However, in reality, some settlement-free peers may attempt to short cut and modify advertisements resulting in you having traffic a bit further than needed. This presentation looks at ways this is done today, some specific examples of this as well as other interesting things you can learn by examining routes received (but maybe not accepted) at all points in a network.

**About the Presenters**

Tom Scholl is a Senior Technical Consultant in the global IP core network design & routing group in AT&T Labs. He works on network design and routing architecture as well as the SBC network integration. Tom has spent his last several years at what was SBC and prior to that American. When not working, Tom can be found on IRC discussing routing, networking hardware and the MBNAF protocol.

Aman Shaikh is a member of Network Performance and Engineering Department at AT&T Research where he works on IP route monitoring and several projects related to IP routing. His general research interests lie in the areas of IP routing and network management. Aman obtained his Ph.D. and M.S. from University of California, Santa Cruz. His home page can be found at [http://www.research.att.com/~shaikh](http://www.research.att.com/~shaikh).

Richard Steunenberg is the Co-Founder of Layer Communications, where he currently serves as Chief Technical Officer and devotes a significant amount of time to the strategic management of peering relationships. Previously, he served as Sr. Network Engineer for several large NAPs, and was the Sr. Software Engineer responsible for developing optimized routing technologies at netVista, Inc.
Network defenses

- Deep packet inspection at line-rate
  - Stop SYN DoS. Use IPS-like features
- Secure Shell, Public key crypto
  - Stop using cleartext. Portknocking, GPG, OTR
- Protecting the infrastructure
  - Reflection and amplification attacks

http://www.nanog.org/mtg-0405/mcdowell.html
Advanced network defenses

- Route filtering, SIDR, PGP-Whois
  - https://Prefix.PCH.Net
  - IETF SIDR WG
  - DHS SPRI
- Proxy scanners, pedantic perimeters
  - Separate DNS servers from infrastructure
  - Never allow outbound SYN from DMZ
  - Perform Firewall Differential Analysis
- SSL (DHE), Cisco port-security + DAI
  - Stop MITM attacks
  - Enforce local DNS/DHCP (not over WAN)

Secure Inter-Domain Routing (sidr)

Last Modified: 2007-02-20
Additional information is available at tools.ietf.org/wg/sidr

Chair(s):
- Sandra Murphy <sandy@tislabs.com>
- Geoff Huston <gfh@apnic.net>

Routing Area Director(s):
- Ross Callon <rcallon@juniper.net>
- Bill Forner <forner@research.att.com>

Routing Area Advisor:
- Ross Callon <rcallon@juniper.net>

Technical Advisor(s):
- Steven Bellovin <snb@cs.columbia.edu>

Mailing Lists:
General Discussion: sidr@ietf.org
To Subscribe: sidr-request@ietf.org
Web vulns: extra borken stuff

Steal Browser History Without JavaScript

Well, the server is back up and running (big thanks to id - during our upgrade there was a drive failure causing us to have to switch machines), and to celebrate I didn’t want to come back with a boring post that would make you question why you read this site. So instead I decided to play around with some CSS tricks - bare with me for a minute. I don’t know why, but I really think CSS is going to get worse over time. Anyway, as I was poking around I happened across one of the missing pieces of the puzzle to solve a simple problem in using CSS to hack - the lack of conditional logic.

Keeping this in mind it would be great if you could create a form of conditional logic in CSS. Well I finally figured out a way. Using a hybrid of a visited and display: attribute you can detect that the user has visited a page and more importantly perform an action based on that fact. The actions are somewhat limited if you can’t use JavaScript, however, one action is enough. The reasoning being, when something is set to display:none it will actually cause the HTML tag that it references to not render. Setting the background image attribute for the visible tag to use a URL of a logging CGI script allows you to send a request to a remote webserver based on the conditional logic as mentioned above.

Now, the only lacking part is the state management, and that can easily be tied together using a unique cookie, and/or an IP address in the QUERY_STRING or anything else you want to use to identify the user. In this way, the remote website can steal history information from the user without ever once using JavaScript, or any client side programming. Click here for a proof of concept of the CSS history theft without using JavaScript. This works nearly instantly, so it is far better than the JavaScript-less intranet hacking and pdp’s version of the JavaScript CSS history hack in terms of speed. The only latency is the time it takes your browser to request the images associated with each URL you’ve visited - which is nearly instant since I don’t return any data (and thanks to browser threading). The other nice thing about this is that it works beautifully in both Internet Explorer 7.0 and Firefox 2.0.0.2 (although it doesn’t work in Opera 9.22).

So now we’ve eliminated the JavaScript pre-requisite from Intranet port scanning, cross site request forgeries, session riding and of course CSS history hacking. The only thing we can’t yet do without JavaScript is read cross domain (and I stress the word yet). What else is left? I don’t mean to sound ho-hum about this, but really, what else do we have to do? Are there any nay-sayers left?

• No order for HTTP headers (except method + version)
• Forms and cookies can be huge, unvalidated
Advanced web vulns: trust\textsuperscript{2}

- mhtml vulnerability: own the whole browser
- Firefox pop-up blocker reads local files
- Exponential Cross-Site Scripting (XSS \textsuperscript{n})
- Ajax and XHR now allow breaking of same-origin. … so does any external JS/DHTML
- Flash does, too, if not configured properly. If configured properly: flash can be good!
- Anti-DNS Pinning (via JS, Flash, and Java applet)
- Cross-site cooking, Homograph attack
- HTTP request smuggling (and splitting, et al)
- HTML and Javascript network/port scanning
Web+Network attacks

• IP address-based authentication
  – Spoofing and MITM (Wireless) break this
  – BGP and MPLS break this
  – Proxies break this
  – Layer One Session Hijacking

• XSS Shell
  – Combination of XSS proxy, History/cache enumeration, JS network/port scanning, clipboard grabbing, and even instant messaging!

Solutions Worthy of Paranola

There is hope, or rather, there are gruesome hacks, that can bring the splendor of seamless cross-browser XMLHttpRequests to your developer palette. The three methods currently in vogue are:

1. Application proxies. Write an application in your favorite programming language that sits on your server, responds to XMLHttpRequests from users, makes the web service call, and sends the data back to users.

2. Apache proxy. Adjust your Apache web server configuration so that XMLHttpRequests can be invisibly re-routed from your server to the target web service domain.

3. Script tag hack with application proxy (doesn’t use XMLHttpRequest at all). Use the HTML script tag to make a request to an application proxy (see #1 above) that returns your data wrapped in JavaScript. This approach is also known as On-Demand JavaScript.

The basic idea of all three hacks is the same: fool your user’s web browser into thinking that the data is coming from the same domain as the web page.

A word of caution here: there is a good reason why XMLHttpRequests are restricted. Allowing them to freely access any domain from within any page opens up users to potential security exploitation. Not surprisingly, these three hacks, which offload the request to your web server, potentially threaten to disparage your web server’s identity, if not its contents. Caution is advised before deploying them.

• WS Proxy Injection
Web+network attack examples

• Using Tor with .edu exit nodes over the Internet
  – StrictExitNodes 1
  – ExitNodes a.edu, b.edu, c.edu

• HTTP Referer checks stop proxies? Spoof the headers!

• My 0-days that you can bring home to mom
  – ProQuest: vulnerable
  – LexisNexis: vulnerable
  – ACM / IEEE: vulnerable
Credit card processors
- Need to create accounts en-masse
- Allow partners by IP prefix

DISA.MIL publications
- Uses reverse DNS lookup checks
- I’m a .mil, let me in!
- DSN directory

Vulnerable? Very likely
Others - Dialog, Hoover’s, InfoTrac, LawTel, and PayScale
Web+Network defenses

- **IP address-based authentication**
  - Don’t do that then! Check your firewalls! Use individual login accounts with passwords!
  - Check your IIS -> Directory Security tab -> IP address-based access control
    - AOL Proxies, Corporate NAT makes this superbad!

- **XSS and CSRF protections**
  - XSS for coders: Input validation with frameworks
  - XSS for operators: Output filtering with WAF’s
  - CSRF: No GET’s! POST with nonce (e.g. Viewstate)
Reflections on trusting trust

My Reflections on Trust

I was a young IT system administrator when Kim Jong-Un started in 1984 years. Reflections on trust. A recent trip to a data center in the Philippines, I was working on the old

evaluation of Unencrypted Windows with the new follow at the National Computer Security Center and conducting some research in the growing Rainbow Banner Book. The theme in ongoing

studies over the lifetime of themes such as "Crypto-based security" and "Unilateral trust" in the "Orange Book" (a small word that was updated in the seven years), and I was pegged into the

wars of building and "trusted computing" instead of "access computing."

We talked a lot about "how much trust do I get for how much analysis and testing" and "how much trust do I need for certain scenarios, like allowing a computer to automatically control a data flow. Top

Secret/Scarlet. These kinds of schemes would be impossible to manage in real life, even for small violations. It was immediately obvious that

getting up to, say, 1% of what you gain over your countryside you trust so you would want to trust the most you trust in a trustee. It’s important to recognize the risks.

When you have trusted old computers, you can immediately determine that

the most sensible and consistent, and the consistency of consistent human factors, you really had no idea how even the most serious consequences would be possible.

It’s very important to recognize that the government, whether you’re aware, much less to pick up on the test environment because mobile computers were not really a problem yet.

The government, whether you’re aware, much less to pick up on the test environment because mobile computers were not really a problem yet.

Why should you care about my early warnings? Well, because I think that trust is still a major problem today and it’s causing everyone a bunch of money that could be put into real long-term solutions.

As I talk with my operations teams these days, I’m seeing a subtle shift in their thinking. They’re thinking more and more about endpoints (end-user systems, servers, IDS and stuff like that), but not for

short-term policy. They seem to be thinking that more than halfway they really believe it can be fixed if they do something right, and most of them have security groups. They have to do all the right things, and they’re working on

fixing the problem, but I really believe that the government, whether you’re aware, much less to pick up on the test environment because mobile computers were not really a problem yet.

The problem is that too many organizations throw out requirements documents from product managers and just start over. Is it only because the requirements are so bad, or is it also

become that broken and just don’t trust the engineers to know what they’re talking about?

And why do many managers try to sell their investment organizations on new security applications, instead of simply what the current state of security is, and just simply trust the development in because

of business objectives, or do they just not understand the creative process around?

And so on.

What an enormous amount of waste cycles that could be saved to make organizational good.

We ease younger to the point where we can validate our trust computers. But, can’t we get to the point where we can trust each other?

• You can’t trust compilers
• You can’t trust people or groups
• You can’t trust networks
• You can’t trust browsers
• You can allow individuals access and make them accountable (sudo vs. su)

– Enforce a password policy
– Never make group accounts
– Watch accounts for activity, monitor policy
– Close or lock-out old/unused accounts
My incident response strategy

• Strike-back
  – You want personal information? Yes, have some (unload tons)
  – Use SQL injection, create large lists with Fakenamegenerator.com

• Lock-down
  – Better to gather intelligence than lock-out usually
  – Allow user to input additional account information for authorization and then force them to change their password

• Trace-back
  – Check web/firewall/IPS/load-balancer logs, SIM/SIEM, NSM data
  – Notify the user/users/groups/orgs. Have them check their logs, too

• Take-down
  – Never operate alone. Use a service e.g. CastleCops PIRT
The future of attacks

- Transitive attacks
  - Web 2.0 (Blogger, Technorati, Sitemeter, Flickr, Zoomr, Feedburner, Newsgator, del.icio.us, et al)
    - Blogs
    - RSS
  - IP address-based authentication / trust relationships
    - Network A – Network B – WebApp Y – Database Z
  - Combination of the above

- Targeted attacks
  - Profiling, stalking (e.g. Google hacking, social network sites, WiFi stalking, AOLstalker.com)
  - Stealing search engine queries from orgs
Example One: Project Camwhore

- This is a targeted attack

4chan
- Front Page
- [Remove Frames]
- [Show Directories]
- [Show WorkSafe Only]

Text Boards
- Anime & Manga
- Animals & Nature
- Rant
- Anime/Cute
- Cosplay & EGL
- Food & Cooking
- Cute/Male
- Comics & Cartoons
- Hentai/Alternative
- Ectoplasm
- Technology
- Animated GIF
- Hentai
- High Resolution
- Weapons
- Mecha

4Chan: Togi-chan
- Encyclopedia Dramatica / Camwhore
- Yahoo Profile -> Deviant Art -> Myspace
Example One: Owning Myspace

Thursday, 25 January 2007

Myspace Allegedly Kills Computer Security Website
Computer security guru Fyodor (pictured) reports waking up yesterday to find his website SecLists.org essentially removed from the web by his domain registrar, GoDaddy. After a bunch of phone calls to GoDaddy, he eventually got them to explain why: Because MySpace asked them too.

SecLists provides public archives of over a dozen computer security mailing lists, including BugTraq and Full Disclosure. MySpace was apparently unhappy with a post that crossed Full Disclosure earlier this month, in which the author attached the spoils of a phishing attack against MySpace users, consisting of 56,000 user names and passwords.

• Myspace hates hackers
  – Samy busted
  – Seclists.org shutdown by GoDaddy
• Myspace loves hackers
  – Allow XSS everywhere (allowing user-submitted HTML is bad, mmkay?)
  – Requires Javascript everywhere
• Myspace is confused
Example One: What’s possible?

• Google cache gave me her Myspace page

• Logic flaw in Myspace allowed me to post on her private profile (we’re not even Myspace friends!)

• Earlier information (ED diff) gave up where she is and how to find her irl

— Currently, Togi-chan works at CinnZeo, at Metrocenter Mall in Phoenix, Arizona
Example 2: Project NSA Call DB

- This attack is at first targeted and then [theoretically] becomes a transitive attack

- Another woman
- She makes news for getting arrested
- How to find her?

Fan hacks into cell phone data for Linkin Park's lead singer, threatens wife, feds say

The Associated Press

ALBUQUERQUE, New Mexico: A woman is accused of using a computer at a national laboratory to hack into a cell phone company’s Web site to get a number for Chester Bennington, lead singer of the Grammy-winning rock group Linkin Park.

According to an affidavit filed by the Department of Defense Inspector General, Devon Townsend, 27, obtained copies of Bennington’s cell phone bill, the phone numbers he called and digital pictures taken with
Example 2: Myspace again!

• Search for:
  – Female
  – Age 27
  – In Albuquerque, New Mexico
  – Works for Sandia National Labs
  – Likes Linkin Park

• ONE RESULT

http://profile.myspace.com/index.cfm?fuseaction=user.viewprofile&friendID=36977809
Example 2: DoD Response

- A few more words about incident response
  - Transitive attack could be as simple as: Myspace – SNL browser – NSA DB
- Need-to-know basis
- A trap?
  - CIA involvement on Facebook

CIA Gets in Your Face(book)

By Chaddus Bruce | Also by this reporter 02:00 AM Jan, 24, 2007

If you're a Facebook member, a career as a government spook is only a click away.

Since December 2006, the Central Intelligence Agency has been using Facebook.com, the popular social networking site, to recruit potential employees into its National Clandestine Service. It marks the first time the CIA has ventured into social networking to hire new personnel.

The CIA's Facebook page (login required) provides an overview of what the NCS is looking for in a recruit, along with a 30-second promotional YouTube video aimed at potential college-aged applicants. U.S. citizens with a GPA above 3.0 can apply.
Summary

- Same-origin policy doesn’t protect you
- Transitive attacks
- Targeted attacks

- There is a lot more out there than what I’ve talked about. The webapp security landscape is constantly and consistently getting worse and worse by the day
References

- http://www.theta44.org/karma/
Questions?