Making HTTPS and Anonymity Networks Slightly More Secure

(Or: How I'm Using My Botball Skill Set in the Privacy/Security Field)

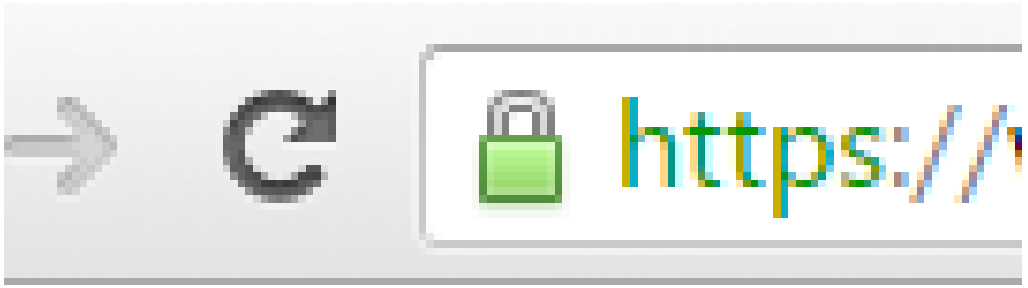
Jeremy Rand
Lead Application Engineer, The Namecoin Project
(Alumni, Norman Advanced Robotics / Team SNARC)

A little bit about me…

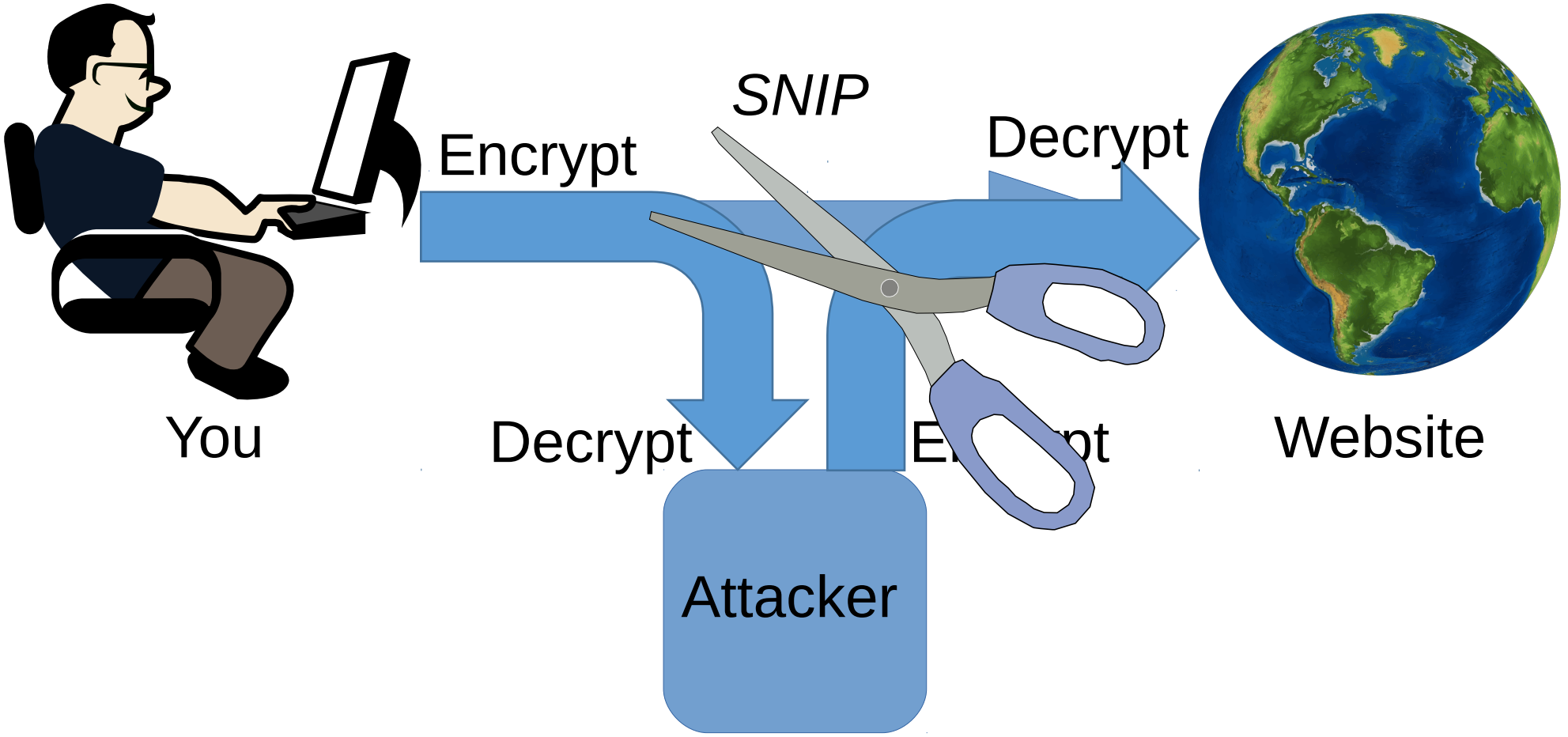
- Founder+Leader of Team SNARC (Competed in KIPR Aerial and KIPR Open 2011-2015).
- Alumni of Norman Advanced Robotics (Class of 2011).
- Mentored Alcott and Whittier Middle Schools 2011-2015.
- Presented at GCER on hacking the XBC, CBC, Link, AR.Drone, and Create (2008-2015).
- Interested in the intersection of technology and human rights.

HTTPS: what does it do?

- When you visit an HTTPS website, that means it's supposed to be secure.
- What does “secure” mean?
- It’s encrypted, but that’s not all that happens.



What you actually expect:



This is called a *Man-in-the-Middle (MITM) attack*

- For encryption to be secure, you need to **authenticate** that the website server you're talking to is actually whom you think it is.
- Standard solution is to introduce **Certificate Authorities (CA's)**.

Certificate Authorities (CA’s)

- Certificate Authorities are corporations that sign **certificates**, which are sort of like ID cards for authenticating websites.
- If you trust a CA, then you can trust all the websites that they've signed a certificate for.
- Over 1000 CA’s are trusted by your web browser.

Wait a minute, this sounds fragile….

- Yep. Very fragile.
- If any of the 1000+ CA's that you trust, makes a mistake…
  - They could issue a false certificate to an attacker, that allows them to do a MITM attack.
- But surely, this hasn't happened, has it?

Yes, it's happened.

- In July 2011, the CA DigiNotar was compromised.
  - Possibly by an Iranian intelligence agency.
  - The attackers got away with fake signatures for impersonating the CIA, MI6, Facebook, Microsoft, Skype, Twitter, WordPress, Mozilla, and hundreds of other targets.
  - DigiNotar didn't even notice for over a month.

More CA Fails…

- The CA WoSign issued a certificate in 2016 for github.com…
  - … to a random guy who only proved that he had an account on GitHub.

Namecoin: like a CA, but no trust required

- Namecoin is very much like Bitcoin.
- But while Bitcoin transactions move money around…
  - Namecoin transactions register and update website addresses.
  - Namecoin website addresses end in .bit
- Namecoin addresses are difficult to impersonate, for the same reasons that bitcoins are difficult to steal.

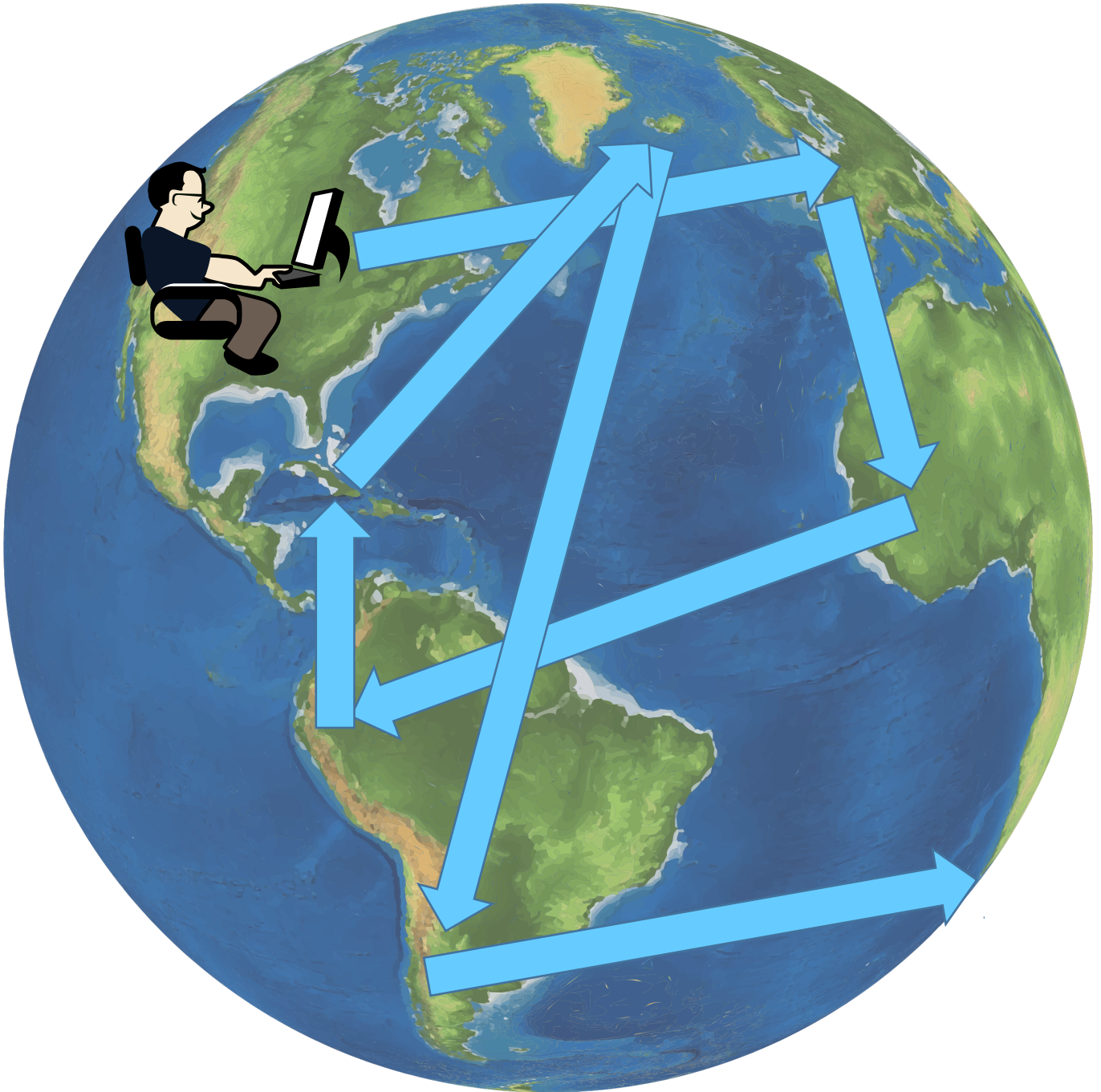
Namecoin improves HTTPS security

- If you register a Namecoin website address, you can control which HTTPS certificate is allowed for it.
  - No trusting CA’s required.

The Tor Anonymity Network

- HTTPS keeps the **content** of your Internet traffic secret.
- But it doesn’t hide **which websites** you’re visiting.
  - Knowing which websites you visit can reveal a lot of private information about you.
- To solve that, you need Tor.

How Tor makes you anonymous



Tor has a usability problem

- A website address that's hosted with Tor looks like this:
  - ~~https://idnxcnkne4qt76tg.onion~~
  - https://odmmeotgcfx65l5hn6ejkaruvai222vs7o7tmtllszqk5xbysola.onion
- Namecoin addresses can point to Tor addresses too.
  - So you won't have to deal with impossible-to-remember Tor addresses if you use Namecoin.

How is this similar to Botball?

- It’s actually very similar.

Reverse-Engineering in Botball

- The controllers and software provided in Botball don’t necessarily do what you want.
  - It might also be undocumented.
- You might experimentally reverse-engineer things in order to make them do what you want.
  - This was the basic formula for all of the Botball hacking papers I wrote.

Reverse-Engineering in Namecoin

- The HTTPS implementations in web browsers also don’t do what I wanted.
  - And the documentation was minimal.
- I had to reverse-engineer parts of the Windows HTTPS implementation in order to make it work with Namecoin.
  - This felt just like I was back in Botball reverse-engineering the CBC.

Questioning Assumptions about Adversaries in Botball

- A poorly kept secret about Botball D.E.: the most well-built and well-programmed robots don’t always win.
- The most critical skill in Botball D.E. is accurately guessing what other teams will try to do.
  - This can let you block your opponent from scoring.
  - This can also let you score reliably even when your opponent is trying to block you.

Questioning Assumptions about Adversaries in Namecoin

- Questioning assumptions is a huge part of security engineering.
- Example: if you get web browsers to accept Namecoin HTTPS certificates, did you remember to make sure that CA's can't issue certificates for Namecoin websites?
  - The former doesn't imply the latter.
- Example: if you get web browsers to block connections to Namecoin sites with the wrong certificate, did you remember to make sure that the blocking happens **before** the browser tries to send login data to the website?
  - If not, then a MITM attacker can steal logins.

Minimizing Attack Surface in Botball

- Avoid unnecessary complexity! (AKA the KISS Principle.)
- Security by isolation: use a blocker robot to “isolate” the robot that scores most of your points from the other team’s robots.
- Avoid known past failure modes: keep track of what strategies didn’t work well (for your team or other teams) and avoid them in the future.

Minimizing Attack Surface in Namecoin

- Avoid unnecessary complexity! A lot of our engineering effort is spent on complexity reduction.
- Security by isolation: keep sensitive code/data separated from code that an adversary can interact with.
- Avoid known past failure modes: memory safety bugs are historically very common in C code; replacing C with safer languages like Go and Rust tends to make things more secure.

International Collaboration in Botball

- Botball teams often form alliances.
  - Swapping code.
  - Sharing tips.
  - Sharing intel.
  - Co-writing GCER papers.
- Often Botball teams in different states or different countries will collaborate.

International Collaboration in Namecoin

- Namecoin developer team scattered across countries.
  - Developers in U.S. (Oklahoma, Texas, Washington state, Connecticut), Switzerland, Germany, U.K., Sweden, Canada.
  - Former developers in France and Russia.
- Several developers operate under pseudonyms.
  - Some don’t disclose what country they’re in.
- Development is entirely coordinated online.
- We collaborate with other project teams.

Why you might want to join Namecoin

- Open-source software development experience looks great on a resume or college application.
- Making the world a better place for human rights (e.g. privacy) is good too.
- The blockchain technology used in Bitcoin and Namecoin has a lot of industry attention these days.

Do you know, or want to learn, any of these?

- Python
- C++
- Go
- Java
- Javascript
- PHP
- Qt GUI's
- PyQt GUI's
- Usability testing
- Documentation
- Packaging (any OS)
- Browser extensions
- Android apps
- DNS
- TLS
- Bitcoin
- Anonymity
- Sandboxing
- Basic applied cryptography
- Unit / integration testing
- Static analysis

jeremy@namecoin.org https://www.namecoin.org